THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Informatique et Génie Industriel

Présentée et soutenue par :

M. DA XU

le mercredi 15 novembre 2017

Titre :

Contribution to the elaboration of a decision support system based on modular ontologies for ecological labelling

---

Ecole doctorale :

Systèmes (Systèmes)

Unité de recherche :

Laboratoire de Génie de Productions de l'ENIT (E.N.I.T-L.G.P.)

Directeur(s) de Thèse :

M. BERNARD ARCHIMEDE

M. MOHAMED HEDI KARRAY

Rapporteurs :

M. DIMITRIOS KIRITSIS, ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

Mme MARIE-HELENE ABEL, UNIVERSITE DE TECHNOLOGIE DE COMPIEGNE

Membre(s) du jury :

M. DAVID CHEN, UNIVERSITÉ DE BORDEAUX, Président

Mme HAJER BAAZAOUI, UNIVERSITE DE MANOUBA, Membre

Mme MOUNIRA HARZALLAH, UNIVERSITE DE NANTES, Membre

Acknowledgements

This three years' study and work is an extraordinary experience for my journey to France that started five years ago. I appreciate that I have met so many outstanding people and interesting things that beautiful stories and memories can be recollected in the rest of my life.

Firstly, I want to thank my supervisor Mr. Bernard ARCHIMEDE and Mr. Mohamed Hedi KARRAY. They have given me a lot of guidance. I was touched and influenced by the passion and diligence of Hedi. The wisdom and experience of Bernard have also inspired me a lot. Thanks to Bernard, I got chance to be the candidate for the thesis and finally was accepted. And thanks to both of them, I managed to be founded as an intern to visit École polytechnique de Montréal for three months. My special thanks to Michel GAGNON and Amal ZOUAQ. During my stay in Montréal, they showed much kindness and gave me useful scientific guidance.

I would like to thank all my dear colleagues for their generous help and care. In particular, thank Chayma KHALFAOUI and Linda ELMHADBI for their review and correction on my English writing; thank Thierry LOUGE for those inspiring discussion. It is my honor and pleasure to work with them.

Thanks to my family and friends. Their regards and encouragement have always been the source of my strength and perseverance.

Abstract

With the rising concern of sustainability and environmental performance, eco-labeled products and services are becoming more and more popular. In addition to the financial costs, the long and complex process of eco-labeling sometimes demotivates manufacturers and service providers to be certificated. In this research work, we have proposed a decision support process and implemented a decision support platform aiming at further improvement and acceleration of the eco-labeling process in order to democratize a broader application and certification of eco-labels. The decision support platform is based on a comprehensive knowledge base composed of various domain ontologies that are constructed according to official eco-label criteria documentation. Traditional knowledge base in relational data model is low interoperable, lack of inference support and difficult to be reused. In our research, the knowledge base composed of interconnected ontologies modules covers various products and services, and allows reasoning and semantic querying. A domain-centric modularization scheme about EU Eco-label laundry detergent product criteria is introduced as an application case. This modularization scheme separates the entity knowledge and rule knowledge so that the ontology modules can be reused easily in other domains. We explore a reasoning methodology based on inference with SWRL (Semantic Web Rule Language) rules which allows decision making with explanation. Through standard RDF (Resource Description Framework) and OWL (Web Ontology Language) ontology query interface, the assets of the decision support platform will stimulate domain knowledge sharing and can be applied into other application. In order to foster the reuse of ontology modules, we also proposed a user-centric approach for federate contextual ontologies (mapping and integration). This approach will create an ontology federation by a contextual configuration that avoid the "OWL:imports" disadvantages. Instead of putting mapping or new semantics in ontology modules, our approach will conserve the extra contextual information separately without impacting original ontologies or without importing all ontologies' concepts. By introducing this contextualization, it becomes easier to support more expressive semantics in term of ontology integration itself, then it will also facilitate application agents to access and reuse ontologies. To realize this approach, we elaborate a new plug-in for the Protégé ontology editor.

Key words: Modular ontologies, Decision support system, Eco-labels, Semantic Web, Ecological certification.

Résumé

L'usine du futur et les performances environnementales sont de nos jours au cœur des préoccupations. Les produits et services éco-labellisés sont de plus en plus populaires. En plus des coûts financiers engendrés, les processus d'éco-labellisation sont longs et complexes, ce qui démotive parfois les fabricants et les fournisseurs de services à demander des certifications. Dans ce contexte, ce travail de recherche, propose une démarche et une plateforme d'aide à la décision visant à améliorer et à accélérer ce processus afin de démocratiser l'accès à la certification écologique. Les bases de connaissances traditionnelles étant généralement peu interopérables, difficiles à être réutilisées et ne supportant pas les inférences, la plate-forme proposée repose sur une base de connaissances composée de diverses ontologies de domaine construites selon la documentation officielle européenne sur les écolabels. Cette base est composée de modules d'ontologies interconnectées couvrant divers produits et services. Elle permet d'automatiser le raisonnement sur ces connaissances et de les interroger en tenant compte de la sémantique. Un schéma de modularisation orienté suivant le domaine et la catégorie du produit, et portant sur les critères d'écolabels européens des produits détergents est utilisé comme cas d'application. Afin de permettre une réutilisation aisée des modules d'ontologie pour différents groupes de produits, ce schéma de modularisation fait la distinction entre la connaissance de base du domaine et les connaissances variables concernant les critères de labélisation de chaque groupe. La méthode de raisonnement utilisée exploite les mécanismes d'inférence sur des règles SWRL, et fournit des résultats argumentés pour l'aide à la décision. La modélisation adoptée pour la représentation des connaissances n'est pas uniquement dédiée à la plateforme proposée. Elle permet également une exploitation des connaissances via des outils du Web sémantique. Afin de favoriser la réutilisation des modules d'ontologie, une approche de contextualisation pour la fédération d'ontologies a été proposée. Elle permet de pallier les inconvénients de "OWL: imports". Contrairement aux approches existantes, où il est nécessaire de réaliser soit un mapping, soit d'ajouter des relations sémantiques modifiant les modules d'ontologies de base, notre approche n'affecte pas et ne nécessite pas l'importation de tous les concepts de ces ontologies. Pour faciliter la mise en œuvre de cette approche, nous proposons un nouveau plug-in pour l'éditeur d'ontologie « Protégé ».

Mots-clés: Ontologies modulaires, Système d'aide à la décision, Labellisation écologique, écolabels, Web sémantique.

Long Résumé

Remarque: Ceci est un résumé succinct et vulgarisé en français du manuscrit rédigé en anglais. Nous avons essayé de fournir ici, pour les lecteurs francophones, une idée générale sur le contexte, les problématiques, les verrous scientifiques traités dans cette thèse et les contributions effectuées lors de ce travail. Vu la difficulté de l'exercice de résumer ce long travail de recherche, certains détails, arguments et références bibliographiques ne sont pas présents dans ce résumé mais ils ont été bien présentés dans la partie rédigée en anglais de ce manuscrit. En vous souhaitant une bonne lecture.

Contexte Général

De nos jours, les challenges environnementaux pour une économie et une société durables préoccupent de plus en plus les autorités publiques, les ONG et les citoyens. L'objectif principal consiste à atteindre un certain équilibre entre la durabilité écologique, le développement de l'économie et la qualité de vie. Dans ce contexte où plusieurs efforts visent à protéger l'environnement et l'équilibre écologique, la politique gouvernementale est l'une des forces dominantes. Outre toutes sortes de règlementations obligatoires, certains moyens volontaires sont également en cours d'élaboration et impactent favorablement les orientations réglementaires sur le marché. L'éco-labélisation faisant partie de cette catégorie de moyens.

En effet, l'éco-labélisation devient de plus en plus populaire dans le monde industriel. Des centaines d'écolabels de différents catalogues de produits et de services sont fonctionnels dans le monde entier. D'une manière générale, les labels écologiques sont une sorte de certifications assignées à des produits qui utilisent une approche tenant compte de tous les impacts environnementaux générés (matières premières, énergie, conservation de la biodiversité, pollution de l'eau, de l'air, du sol, des déchets, du bruit, etc.) et de toutes les phases du cycle de vie du produit (de l'extraction des matières premières, passant par la fabrication, la distribution et la réutilisation et jusqu'à la fin de la vie du produit) selon les exigences définies dans les spécifications environnementales. Le label écologique identifie globalement la préférence environnementale d'un produit ou d'un service dans une catégorie de produit / service spécifique. Le processus d'éco-labélisation garantit la qualité d'utilisation du produit et limite l'impact sur l'environnement. Ces processus sont de plus en plus communs et décisifs dans notre consommation quotidienne et dans d'autres activités commerciales.

Ce processus de certification écologique est avantageux pour les différentes parties prenantes sous différents angles. Outre la sécurité, la santé et la qualité, un certain nombre de consommateurs lambdas s'attendent de plus en plus à ce que le produit ou le service consommé respecte l'environnement. L'écolabel est un bon média pour communiquer et transmettre ces informations et ces indications à ces consommateurs. En ce qui concerne le contenu de l'éco-labélisation, des graphiques et des textes spécifiques sont généralement imprimés sur les produits afin de mettre en évidence les compétences environnementales ainsi que d'autres qualités. Les consommateurs peuvent donc identifier ces produits qui correspondent le mieux à leur bonne volonté par rapport à l'environnement. Les étiquettes écologiques influencent le choix des consommateurs et stimulent la consommation « verte ». Bien sûr, il existe tout de même des soucis en ce qui concerne le modèle et le contenu des labels écologiques qui deviennent trop riches et spécifiques au domaine de façon que les consommateurs se trouvent souvent submergés par ces informations d'écolabels. De plus, en raison du nombre croissant et de l'abus de toutes sortes d'écolabel, la qualité et la crédibilité de certaines étiquettes (labels) continuent de diminuer. Cependant, les étiquettes écologiques restent toujours de bons outils pour guider le pouvoir d'achat et pour inciter à une consommation responsable. L'écolabel a toujours été un sujet intéressant de l'économie, de la société, du droit, de l'éthique, etc.

Pour les producteurs et les fournisseurs de services, l'éco-labélisation peut être une méthode pour augmenter la compétitivité de leurs produits ou services. Il a été prouvé que la valeur ajoutée représentée par certains écolabels est attrayante pour certains consommateurs. D'autant plus que certains groupes d'étiquettes écologiques ne sont pas seulement liés à la performance environnementale mais aussi à la sécurité. D'autres performances d'utilité sont également inclus comme par exemple dans le cas des écolabels Européen. Dans une telle situation, les produits éco-labélisés peuvent pousser les utilisateurs à croire qu'ils sont meilleurs que les produits non labélisés. Du point de vue du producteur et de l'entreprise, l'éco-labélisation (la certification écologique) peut être une méthode rentable pour améliorer la visibilité du produit sur le marché. En d'autres termes, l'écolabel est en quelque sorte un autre type de publicité et de mise en valeur du produit.

Il est à noter qu'une différence importante entre l'éco-labélisation et d'autres réglementations ou normes, est que l'éco-labélisation est principalement une démarche volontaire. Sur la base de la réglementation standard ou de la norme, l'éco-labélisation offre plus d'espace de concurrence et de flexibilité marketing. Dans une certaine mesure, il stimule la « production verte » et essaie de diriger l'ensemble de l'industrie vers une direction plus efficace et plus propre.

Cependant, toutes les entreprises ne connaissent pas les écolabels. Ils ne connaissent pas non plus le processus de son obtention. Face à tant d'écolabels utilisés dans le monde entier, les entrepreneurs se sentent parfois perdus. En outre, la plupart des labels écologiques ne sont pas gratuits, des dépenses considérables doivent être payées pour candidater à l'obtention d'un écolabel et généralement ce processus de labélisation est long. Ceci pourrait entraîner des difficultés financières pour les PME (petites et moyennes entreprises) qui doivent faire appel parfois à des consultants extérieurs soit pour un audit pré-candidature ou encore pour le montage entier du dossier de candidature.

Pour les autorités publiques, l'écolabels peut être un outil pour promouvoir l'économie verte et la durabilité d'un point de vue global. Il est facile de voir que si la part de marché des produits éco-étiquetés augmente, l'impact environnemental sera réduit de manière correspondante. Bien qu'il soit important de noter que, malgré la prospérité des produits éco-labélisés, des problèmes et des défis existent. Il existe encore beaucoup d'amélioration pour l'éco-étiquetage. Outre ses avantages, l'éco-labélisation est un processus laborieux, en particulier sa tâche d'évaluation qui complexe et longue. Cette tâche, incluse généralement, concernant les catégories des produits industrialisés, l'analyse des caractéristiques physiques et la composition chimique en plus de l'évaluation des impacts environnementaux et d'autres impacts pour les autorités publiques, l'écolabels peut être un outil pour promouvoir l'économie verte et la durabilité d'un point de vue global. Il est facile de voir que si la part de marché des produits éco-étiquetés augmente, l'impact environnemental sera réduit de manière correspondante. Bien qu'il soit important de noter que, malgré la prospérité des produits éco-labélisés, des problèmes et des défis existent. Il existe encore beaucoup d'amélioration pour l'éco-étiquetage. Outre ses avantages, l'éco-labélisation est un processus laborieux, en particulier sa tâche d'évaluation qui complexe et longue. Cette tâche, incluse généralement, concernant les catégories des produits industrialisés, l'analyse des caractéristiques physiques et la composition chimique en plus de l'évaluation des impacts environnementaux et d'autres impacts. En plus de la complexité industrielle de concevoir des produits respectant les critères environnementaux qui freine les industriels à s'orienter vers ce type de produit, s'ajoute les contraintes de temps et de coût.

Nous pensons qu'un processus d'éco-labélisation fiable, plus rapide et moins complexe pourrait être la clé pour donner une meilleure image des écolabels dans le but de démocratiser l'accès à ce genre de labélisation afin de mieux protéger l'environnement et maintenir un développement plus durable. Les industriels auront moins de fardeau financier et cela attirerait naturellement davantage d'entreprises à étiqueter leurs produits.

Le processus d'éco-labélisation repose essentiellement sur l'expertise humaine ce qui explique le coût engendré. Réduire donc l'effort humain dans ce processus aura un impact positif sur la réduction des coûts. Traditionnellement, l'évaluation du profil du produit à labéliser est menée par de multiples analyses, des réunions et d'édition de rapports entre experts de domaines complémentaires. Peu importe la façon dont les informations sont stockées, nous pensons qu'il doit y avoir une partie du processus qui peut être informatisé et automatisé. De plus, les connaissances des experts doivent être aussi capitaliser et réutiliser. Dans le cadre d'une telle approche informatisée, il serait nécessaire de gérer et traiter et rendre interopérable les connaissances et les données complexes impliquées dans ce processus.

D'autre part, du coté des industriels, ceux derniers ne disposent pas aussi d'outils informatiques disponibles et gratuits permettant vérifier si leurs produits respectent les conditions d'attribution d'écolabels ou non, c'est pour ça ils font appel à l'expertise externe qui coute aussi de l'argent. En cas d'échec de leurs demandes d'éco-labélisation, les industriels n'ont pas d'explications détaillées ou de suggestions de substitution pour l'amélioration de leurs produits. Nous pensons qu'une solution informatisée pour le processus d'éco-labélisation devrait également tenir compte de ces problèmes.

Ainsi, l'objectif métier de ce travail de recherche consiste à proposer une approche informatisée pour aider à améliorer le processus d'étiquetage écologique et à résoudre certains problèmes connexes que ce soit pour l'expert évaluateur ou pour les industriels. Nous essayerons dans ce travail de proposer une approche générique qui pourrait être utile et inspirant pour d'autres tentatives visant à améliorer l'étiquetage écologique ou tout autre système de certification.

Objectifs Scientifiques

Après une étude préliminaire, les principaux objectifs métiers de ce travail de recherche sont résumés comme suit : informatiser le processus d'éco-labélisation pour réduire l'implication de l'expertise humaine afin de l'accélérer et en réduire ses coûts. Par conséquent, fournir un outil auxiliaire pour les experts du domaine ainsi que les industriels pour les aider à évaluer le produit ou le service.

En général, le processus étudié dans ce travail et spécialement un processus d'évaluation multicritère et qui tient compte des différentes phases du cycle de vie des produits / services. Une grande quantité de données et d'informations hétérogènes pourrait être impliquée et manipulée. Pendant l'échange et le traitement des données, des problèmes d'interopérabilité peuvent exister. Nous avons également remarqué que le processus d'étiquetage écologique est essentiellement un problème de décision. Multiples aspects du profil de certains produits doivent être examinés et la décision doit être prise en fonction d'une analyse exhaustive. Il est représentatif en ce qui concerne les difficultés auxquelles l'aide à la décision peut traiter. Dans ce travail nous adopterons donc une approche d'aide à la décision à travers laquelle on arrive à atteindre les objectifs susmentionnés.

Ainsi, le premier objectif scientifique de ce travail est de développer un système d'aide à la décision (DSS) pour l'éco-labélisation.

Le mécanisme de fonctionnement général de cette approche d'aide à la décision est le suivant : tout d'abord, un modèle de profil de produit acceptable (« écolabélisable ») ou « Golden standard » doit être défini ; Ensuite, le profil du produit candidat à la labélisation est comparé au standard. Si le produit candidat satisfait l'exigence de la norme, une décision positive devrait être prise, sinon nous aurons une décision de rejet de la demande d'éco-labélisation. Dans ce cas de rejet de labélisation, l'explication ou l'argumentation sont très importantes pour l'expert, l'industriel et toute autre partie prenante.

Pour mettre en œuvre ce système d'aide à la décision, nous aurons besoins des connaissances de domaine d'éco-labélisation généralement publiées dans des documents officiels contenant les critères (le journal officiel dans le cas des écolabels européens) ainsi que d'autres connaissances potentielles comme par exemple des connaissances concernant les processus de cycle de vie, les procédés de fabrication, etc. Il serait bien que le format de présentation de ces connaissances permettra une certaine capacité de traitement pour la comparaison ou de raisonnement pour faciliter la génération d'explications.

Pour autant que nous le sachions, l'ontologie semble être le meilleur candidat pour représenter la connaissance et pour faire des raisonnements dessus. Selon la littérature, une ontologie est une spécification formelle et explicite d'une conceptualisation partagée. À notre connaissance, l'ontologie est un bon système de représentation des connaissances avec une sémantique unifiée et interopérable. Contrairement aux bases de connaissances traditionnelles, l'ontologie aide à décomposer la complexité de la gestion d'un grand nombre de règles et de contraintes en les organisant dans des hiérarchies et des taxonomies bien structurées. L'utilisation d'ontologie permet une réutilisation efficace de l'information ainsi que l'inférence sémantique et le raisonnement.

De ce fait, dans ce travail de recherche, le système d'aide à la décision (SAD) à développer exploitera une représentation ontologique de connaissances du domaine. C'est-à-dire que le système exploitera une base de connaissances construite à base d'ontologies. En effet, avoir une base de connaissances basée sur l'ontologie couvrant les critères d'éco-labélisation et d'autres connaissances concernant le domaine industriel est une condition préalable à la mise en œuvre du SAD. Par conséquent, la construction de la base de connaissances en ontologies est également un objectif scientifique important de cette recherche.

La construction de ces ontologies nécessite extraction des connaissances à partir de documents complexes sur l'écolabels (texte officiel, article de loi, des figures, des tableaux, des formules, etc.). Nous devons trouver donc une méthode appropriée pour réaliser l'extraction et la représentation de ces connaissances. En plus, on s'attend à ce que la base de connaissances de l'ontologie réalisée puisse être réutilisée par d'autres applications ou encore différents types de catégories de produits. La modularité de l'ontologie ou de l'ontologie modulaire est une bonne solution pour cela. Dans cette recherche, un autre objectif scientifique important serait de trouver une méthode ou un schéma pour mieux gérer et intégrer les ontologies modulaires pour une meilleure réutilisation.

Problématiques et verrous scientifiques

Dans les sections précédentes, nous avons accentué la problématique métier ainsi que les objectifs métiers et scientifiques de ce travail de thèse. Pour réaliser ces objectifs scientifiques, nous devons faire face à des problématiques et verrous scientifiques que nous résumons à travers les questions suivantes :

1. Comment obtenir une représentation appropriée des connaissances sur l'écolabel avec l'ontologie ?

    a. Comment extraire les connaissances à partir de formats complexe (figures, tableaux, articles de lois, etc.) et former une conceptualisation bien structurée ?

    b. Comment décomposer les connaissances en différentes ontologies modulaires ?

    c. Comment traduire les critères d'éco-labélisation en règles d'inférence ?

2. Comment gérer la dynamique (l'évolution) des critères d'éco-labélisation ?

a. Comment gérer la mise à jour des règles d'inférence ?

b. Comment traiter gérer la mise à jour des modules d'ontologies ?

3. Comment définir et formaliser un processus informatisé d'aide à la décision pour l'éco-labélisation ?

   a. Comment mettre en œuvre la fonction d'évaluation de base à travers d'une ontologie et d'un mécanisme de raisonnement ?

   b. Comment générer une explication ou une argumentation appropriée à une décision ?

   c. Comment généraliser le processus de prise de décision dédié à certains produits et l'appliquer à tout autre produit ?

4. Comment mieux réutiliser et intégrer les modules d'ontologies ?

   a. Quels sont les inconvénients du mécanisme de réutilisation et d'importation du langage OWL ? Comment dépasser ces inconvénients et améliorer la réutilisation des ontologies sous OWL ?

   b. Comment intégrer les modules d'ontologies au lieu d'utiliser « owl: imports » ?

   c. Quelles sont les autres solutions pour l'ontologie modulaire ? Quels sont les avantages et les inconvénients de ces méthodes ?

   d. Comment réaliser une réutilisation flexible et partielle ainsi qu'une intégration efficace des modules d'ontologies ?

Méthodologie

La méthodologie de recherche que nous prenons est en corrélation avec la méthodologie d'ingénierie ontologique séquentielle. Dans la première phase de notre travail, le problème principal concerne le développement de l'ontologie. Un état de l'art sur les écolabels et l'écolabel européen (UE), qui correspond à notre cas d'étude, a été réalisé. Nous avons choisi le groupe de produits détergents pour lessive comme cas d'illustration. Après une étude supplémentaire sur le référentiel de connaissances existant et les modules d'ontologie réutilisables, aucun résultat approprié n'a été trouvé. Ensuite, nous avons décidé de développer l'ontologie des critères d'éco-labélisation-UE en se prenant appui sur la documentation officielle. Enfin, une série de tests d'évaluation et de raisonnement a été effectué pour valider l'ontologie. Sur la base des résultats obtenus lors la première phase, un processus d'aide aux décisions basé sur l'ontologie a été proposé et sa mise en œuvre en prototype a été développé. L'objet principal de la deuxième phase concerne la façon d'exploiter l'ontologie dans la prise de décision. Dans la troisième phase de notre recherche, nous nous sommes concentrés en grande partie sur la réutilisation de l'ontologie. Étant donné que nous avons déjà développé l'ontologie des critères du groupe de produits détergents pour lessive, nous avons envisagé de réutiliser les modules développés dans l'ontologie des critères d'autres groupes de produits.

Une méthode d'intégration contextuelle pour l'ontologie modulaire (CIMOn) a été proposée. Nous avons essayé de dépasser les inconvénients existants de la syntaxe du langage OWL en introduisant une contextualisation granitisant l'indépendance des modules ontologiques et en permettant une importation partielle.

Contribution

La contribution fondamentale de notre travail est un processus d'aide à la décision basé sur la base de connaissances formée d'ontologies modulaires. Plus précisément, au lieu de dépendre totalement des efforts humains dans le processus d'éco-labélisation, nous proposons un système d'aide à la décision comme outil auxiliaire pour aider les experts à prendre une décision plus judicieuse en termes de certification des labels écologiques. Le système d'aide à la décision sera un outil puissant pour les experts délivrant les écolabels de l'UE, d'autre part, il peut également être un outil de simulation pour les industriels candidats à l'écolabel. En généralisant les étapes et les tâches, le processus de prise en charge des décisions peut également être appliqué à tous les groupes de produits.

Pour assurer le bon fonctionnement de ce système d'aide à la décision, une base de connaissances composée d'ontologie modulaire intégrée a été construite. La majorité des connaissances sur le domaine de l'écolabels étaient traduites et stockées dans des modules d'ontologies. Pour mieux gérer et réutiliser la base de connaissances de l'ontologie modulaire, d'abord, nous avons introduit un modèle de séparation entre le module d'entité et le module de règles en utilisant owl: imports. Ce modèle fait une distinction entre la connaissance descriptive et objective et la connaissance subjective. Cette séparation est en faveur de la réutilisation et de l'extensibilité de l'ontologie. En utilisant ce schéma de séparation, nous pouvons construire des ontologies modulaires bien structurées. Nous avons également démontré comment faire le raisonnement et l'inférence dans l'ontologie des critères d'éco-labélisation en utilisant le raisonneur Hermit. Vu qu'il y a encore un manque d'application de prise de décision basée sur l'ontologie pour l'éco-labélisation, notre travail peut être considéré comme une référence et une expérience pour des tâches similaires. Nous avons également publié cette ontologie sur GitHub, d'autres chercheurs peuvent y accéder et extraire dont ils ont besoin.

Nous avons proposé une méthode d'intégration contextuelle pour l'ontologie modulaire que nous avons appelé CIMOn pour intégrer et réutiliser différents modules d'ontologie. CIMOn est une méthode pour intégrer les modules ontologiques en introduisant un composant intermédiaire appelé «contexte». En utilisant « contexte », les informations supplémentaires qui ne sont nécessaires que lorsque l'intégration des ontologies se produit, ils peuvent être stockées indépendamment. Ainsi, les modules d'ontologies originales restent intacts et peuvent être réutilisés à des fins ultérieures dans différents contextes au même moment. CIMOn est compatible avec owl: imports, tandis qu'un «filtre» est introduit pour réaliser une importation partielle, ce qui signifie que le contenu de l'ontologie peut être partiellement intégré dans le contexte. Avec cette fonctionnalité nous rendons la réutilisation de l'ontologie plus flexible. Bien qu'il existe plusieurs méthodes pour l'intégration des ontologies modulaires et les ontologies distribuées, la plupart d'entre elles restent dans l'élaboration théorique, peu d'aspects pratiques sont discutés et même moins d'outils sont disponibles. Dans notre travail, en plus de la méthode CIMOn, un plug-in Protégé a été développé. Nous envisageons de partager cet outil après un plan d'amélioration et de tests à grande échelle. Dans une certaine mesure, c'est une autre contribution importante de ce travail.

Structure du manuscrit

Le chapitre 1 concerne le domaine des écolabels et de son état actuel. La portée de l'éco-labélisation est définie. Deux raisons principales de préoccupations des écolabels sont discutées. En raison de la grande quantité de labels écologiques, une inspection complète pour chacun d'eux est impossible. A titre d'illustration pour nos recherches, nous avons choisi l'écolabel Européen. À la fin de ce chapitre, les principales problématiques et les actuels défis de l'éco-labélisation sont identifiés.

La première partie du chapitre 2 porte sur l'état de l'art des systèmes d'aide à la décision en termes d'éco-labélisation et de l'exploitation des ontologies pour la décision. Dans ce chapitre, nous pouvons constater que même si les SAD sont déjà largement utilisés, un nombre très limité de travaux et de recherches sur les SAD appliqués aux écolabels peuvent être trouvés. La plupart des recherches sur l'écolabels concernent l'amélioration des processus, la production verte et l'impact social. Cependant, les systèmes d'aide à la décision exploitants des ontologies ont été évoqué dans certains travaux. Dans ce travail de recherche, nous essayons de combler l'écart afin que le système d'aide à la décision basé sur l'ontologie puisse être exploité dans l'éco-labélisation et d'autres systèmes similaires de labélisation ou de certification. Dans la troisième partie du chapitre 2, nous introduisons le domaine des ontologies et l'ingénierie de l'ontologie. Différents aspects sur l'ontologie et des technologies connexes sont présentés. Dans notre recherche, nous nous intéressons davantage à la modularité des ontologies (ontologies modulaires).

Le chapitre 3 décrit en détail la base de connaissances formée d'ontologies que nous avons développée pour le système d'aide à la décision. Une méthodologie d'ingénierie de l'ontologie a été appliquée. Nous exposons comment les connaissances sur les critères d'écolabels ont été extraites et la façon dont l'ontologie des critères a été construite. Dans cette partie du travail, nous avons appliqué une approche modulaire pour diviser l'ontologie en petits modules afin qu'ils puissent être réutilisés plus facilement. Nous démontrons également comment utiliser le raisonnement pour déduire de nouvelles connaissances ; Comment juger si un produit est compatible avec des critères d'écolabel ou non. Enfin, une évaluation et une analyse de la base de connaissances ont été menées. Nous avons également discuté de la valeur ajoutée et des attentes d'un tel type de base de connaissances dans un contexte étendu.

Le chapitre 4 est en fait une exploration approfondie de ce qui a été proposé lors du chapitre 3. Dans ce chapitre, nous proposons la méthode CIMOn permettant d'améliorer l'efficacité et la flexibilité de la réutilisation et de la modularisation de l'ontologie. Nous verrons que cette nouvelle conceptualisation appelée «contexte» est introduite comme un environnement intermédiaire pour intégrer les modules d'ontologie. Une étude des méthodologies actuelles d'intégration d'ontologie modulaire et distribuée est également présentée. Nous discuterons des caractéristiques et des points faibles de ces méthodes. Ensuite, nous expliquerons en détail comment fonctionne la méthode CIMOn; quels sont ses avantages; comment appliquer ce schéma au développement de l'ontologie ainsi que d'autres aspects de l'ingénierie.

Dans le chapitre 5, nous proposons le processus d'aide à la décision intégrant la base de connaissances ontologique décrite dans le chapitre 3. Comme illustration de l'approche, nous avons choisi le référentiel de critères de produit de détergent-lessive de l'écolabel européen. Une description détaillée des étapes du processus d'aide à la décision et de l'architecture du système est présentée. Nous avons également montré la façon avec laquelle les critères de labélisation étaient traduites en règles SWRL (Langage de règle pour le Web sémantique) et comment appliquer les mécanismes de raisonnement sur ces règles pour obtenir un jugement concernant l'éco-labélisation d'un produit ainsi que les arguments.

Le contenu des chapitres 3, 4 et 5 reflète nos contributions principales dans ce travail. Le chapitre 6 porte sur le développement du prototype et de la mise en œuvre de nos travaux de recherche. Nous avons fourni deux implémentations principales dans ce travail. La première est un plug-in de Protégé utilisé pour l'édition du contexte CIMOn. La deuxième est un prototype de mise en œuvre du processus d'aide à la décision, c'est-à-dire le prototype du SAD pour l'éco-labélisation des produits. Nous expliquerons comment nous avons réalisé ces développements, partant de la conception passant par le codage jusqu'à l'expérimentation technique.

Dans le dernier chapitre, nous allons faire un tour d'horizon sur les objectifs et les contributions réalisées dans ce travail. Nous allons également cerner les limites et les problèmes non résolus encore. Nous discutons aussi les potentiels de réutilisations des principales réalisations de ce travail dans la communauté scientifique.

Contents

List of Figures

List of Tables

Chapter 0

Introduction

0.1 Motivation

Environmental and sustainability are becoming more and more important issues in today's economy and society. People are always looking forward to certain balance between nature sustainability, economy development and life quality. Among all the efforts trying to protect environment and ecological balance, government policy is one of the dominant forces. Besides all kinds of mandatory regulation and laws. Some voluntary means is also undergoing and they also provide good guidance and regulatory functions in the market. Eco-labeling belongs to such "soft" means.

Today, eco-labeling is becoming more and more popular in the world wide. Hundreds of eco-labels of different catalogs covering various products and services are in operation around the world. Generally speaking, eco-labels are certification assigned to products using an approach that takes into account all the generated environmental impacts (raw materials, energy, conservation of biodiversity, pollution of water, air, soil, waste, noise, etc.) as well as all phases of the product life cycle (from extraction of raw materials to end of product life, through manufacturing, distribution and reuse) according to the requirements defined in the environmental specifications. Eco-label identifies overall, proven environmental preference of a product or service within a specific product/service category. Eco-labeling processes guarantee the quality of the product and limits the impact on the environment. These processes are more and more common and decisive in our daily consumption and other business activities. Especially in developed economies, most of which have entered the post-industrial era and the population benefits a higher level of income and life quality.

Eco-labels and eco-labeling are advantageous for different participants from various perspectives. Besides safety, health and good-quality, quite a number of individual consumers expect product or service to be environment-friendly. Eco-label is good media to communicate and transmit such information and indication to these consumers. In terms of labeling content, rich graphics and text are usually printed on the products in order to highlight the environmental competences and other goodness. Consumers can purchase those products that best suit their willingness with regards to environmental performance. Eco-labels do influence the choice of consumers and stimulate the "green consumption". While, there are also worries claiming that the pattern and content of eco-labels are becoming too rich and domain-specific so that consumers often find themselves overwhelmed by these eco-labeling information. Also, due to the rising number and abuse of all kinds of eco-labels, the quality and credibility of certain labels keep decreasing. However, eco-labels are always good tools to guide the purchase and consumption power and eco-labeling has always been an interesting topic related to economics, society, law, ethics, etc.

For producer and service providers, eco-labeling can be a method to increase their product or service's competitiveness. It has been proven that the added-value represented by certain eco-labels are attractive for some consumers. Moreover, certain groups of eco-labels are not only related to environmental performance, safety as well as other utility performance are also included, e.g. EU Eco-label. In such situation, eco-labeled products may have consumers believe that they are of better quality in general sense compared to the other non-labeled products. From the point of view of producer and enterprise, eco-labeling may be a cost-effective method to improve the product's visibility in the market. In other words eco-labeling is somehow another kind of advertising. An important difference between eco-labeling and other regulation or standard is that eco-labeling is mostly voluntary. Based on the basic regulation or standard, eco-labeling provides more competition space and marketing flexibility. To some extent, it stimulates the "green production" and tries to lead the whole industry to more efficient and cleaner direction. Today, however, not all the enterprises are aware of eco-labels. Neither are they familiar with the application process. Faced with so many eco-labels being operated around the world. Entrepreneurs sometimes feel at a loss. Also, most of eco-labels are not free, considerable spending has to be paid for the application and labeling process. This could bring financial difficulty for SME (Small and Medium-sized Enterprises).

For government departments or authorities, eco-labeling can be a tool to promote "greener" economics and sustainability from a global point of view. It's easy to see that if the market share of eco-labeled product gets higher, the environment impact will be reduced correspondingly. While, it must be noted that, in spite of the prosperity of eco-labeled products, problems and challenges exist. There is still much space of improvement for eco-labeling. Besides its benefits and advantages, eco-labeling is laborious work. Eco-labeling, especially its evaluation task, is a complex and time-consuming process. Usually, the physical characteristics and chemical composition have to be analyzed. The emission and other environmental impacts have to be evaluated. During these analysis and evaluation, complex domain knowledge and large amount of heterogeneous data are needed as criteria or reference. Domain experts' opinions have to be taken into account and coordinated. For those complex product or services, actual experiment or inspection on site is needed too.

In our research, we find that faster and better eco-labeling process could the key to achieve better eco-labels and then promote and democratize eco-labeled products in order to better protect the environment and keep ecology sustainable. Because if we can reduce the evaluation time and cost for eco-labeling application, producers and enterprises would have less financial burden and that would naturally attract more enterprises to have their products labeled. If we can accelerate the labeling process, enterprises could be more reactive and flexible with their strategy facing market change.

Particularly, in order to improve the eco-labeling process in terms of evaluation time and cost, firstly, we might as well start by reducing human efforts in the process. Traditionally, the evaluation of product profile is conducted by domain experts' meeting and discussion. No matter how the information is stored, we believe that there must be some part of the process that can be computerized and digitized. Next, in a computerized approach, we have to manage and update the involved complex knowledge and data. If possible, we try to improve the interoperability between different eco-labels. From the point of view of eco-label applicants, they still don't have the possibility to check their product before applying. In case of application failure, applicants don't have explanation or substitution suggestion for their product design. We think that a better eco-labeling process should also take these issues into account. Thus, in our research scope, the initial motivation of our research is try to propose a computerized approach to help to improve eco-labeling process and address some related problematics. We hope our work could be helpful and inspiring to other attempts trying to improve eco-labeling or alike certification systems by any means.

0.2 Objectives

After preliminary study, the main objectives of our research are summarized as following: facilitate and improve eco-labeling by accelerating the process and reducing its cost; computerize eco-labeling process; provide auxiliary tool for domain experts as well as applicants to help them with the product or service's evaluation.

In general sense, eco-labeling process especially the evaluation task needs to take into account different criteria and consider different phases of products/services' life cycle. Large amount of heterogeneous data and information could be involved and manipulated. During data exchange and processing, problems in terms of interoperability may exist. We have also noticed that the eco-labeling process is essentially a decision-making problem. Multiple aspects of certain product profile have to be examined and decision has to be made based on a comprehensive analysis. It is representative with regards to the difficulties that decision support can deal with. In our research scope, we plan to take decision support approach as the candidate solution. So, the premier objective of our work is to develop a decision support system (DSS) for eco-labeling to partly replace domain experts' labor in order to improve eco-labeling in terms of labeling speed and quality.

The basic idea or mechanism behind this decision support approach is actually quite simple. First, an acceptable product profile model or "golden standard" should be set. Then application product profile is compared to the standard. If the application product satisfy the requirement of the standard, positive decision should be made, otherwise we will have negative ones. If the application product fails the evaluation, explanation or argumentation will be very important to the decision maker, applicant and other stakeholders. To implement this decision support system, eco-labeling domain knowledge recorded in official criteria documents and other potential knowledge sources is needed in order to compose such a "standard". It had better to be capable of certain comparison or reasoning ability to facilitate explanation generation. As far as we know, ontology seems the good candidate to represent knowledge and support reasoning. An ontology is a formal, explicit specification of a shared conceptualization [5]. To our knowledge, ontology is a good knowledge representation scheme with unified and inter-operable semantics. Confronting knowledge system, ontology helps to break down the complexity of managing a large number of guidelines and rules by organizing them in well-structured hierarchies and taxonomies. Ontology enables efficient reuse of information as well as semantic inference and reasoning. So in our research work, we will have the decision support system operate an ontology representation of knowledge, i.e. an ontology knowledge base. In other words, an ontology based knowledge base covering eco-labeling criteria and other domain knowledge is a prerequisite for the DSS's implementation. The construction of the ontology knowledge base is also an important objective of this research.

In order to achieve the premier objective of improving eco-labeling, a concrete decision making process is needed, i.e. a method or process describing how application product profile is compared to eco-label's standard should be realized. In this process the knowledge needed for decision making is extracted as ontology from complex eco-labeling criteria documentation of text, figures, tables and formulas, etc. We have to find a proper method to realize the extraction and representation of these knowledge. Moreover, it is expected that the ontology knowledge base can be reused by other applications. Modularity of ontology or modular ontology is a good solution for this. In this research, another important objective is to find a method or scheme to better manage modular ontology in favor of better re-usability.

0.3 Scientific issues

In the previous sections of motivation and objectives, we have introduced that the practical significance of our work is to develop a DSS for eco-labeling process in order to improve eco-labeling in terms of labeling time and cost. To realize this research objectives, we have to deal with some scientific issues or difficulties as summarized in the following questions:

- How to achieve appropriate eco-labeling knowledge representation with ontology?

  - How to extract knowledge from various information format and form a well-structured conceptualization?

  - How to build OWL ontologies in a modularized way?

  - How to build rules?

- How to handle the evolution of eco-labeling criteria?

- How to add, remove or change rules?

- How to deal with the add, remove or change of ontology modules?

- How to define and formulate a computerized decision support process for Eco-labeling?

  - How to implement the core evaluation function by means of ontology and reasoning?

  - How to generate appropriate explanation or argumentation for the decision?

  - How to generalize the decision support process dedicated to certain product and apply it to any other?

- How to better reuse and integrate ontology modules?

  - What are the drawbacks of OWL reuse and imports mechanism? How to overcome these drawbacks and improve re-usability in terms of OWL ontology.

  - How to integrate ontology modules instead of using owl:imports?

  - What are the other solutions for modular ontology? What are the advantage and disadvantage of these methods?

  - How to achieve a more flexible and partly reuse and integration of modular ontology?

0.4 Methodology

Figure 1 briefly illustrates the mind map of the research methodology we have taken in our research. Basically, our research is dedicated in three important aspects of ontology engineering: ontology development, ontology application, and ontology reuse which are unfolded in sequence in our work. This methodology actually correlates with sequential ontology engineering methodology. In the first phase of our work, the main issue is about ontology development. A survey of eco-labeling and EU Eco-labeling was conducted. We chose laundry detergent product group as the illustration case of our research domain. After further survey for existent knowledge repository and reusable ontology modules, no appropriate result was found. Then we decided to develop the criteria ontology according to the criteria document. At last, a brief evaluation and reasoning test was carried out to verify the ontology. Based on the out-come of the first phase, a criteria ontology based decision support process was proposed and its prototype system was developed. The main topic of the second phase is about how to exploit ontology in decision support application. In the third phase of our research, we focused largely on ontology reuse. Since we had already developed the criteria ontology of laundry detergent product group, we considered reusing the existent modules into other product groups' criteria ontology. A **Contextual Integration for Modular Ontology (CIMOn)** method was proposed. We tried to overcome the drawbacks of existent OWL ontology syntax by introducing independent "context" and allowing partly importing.

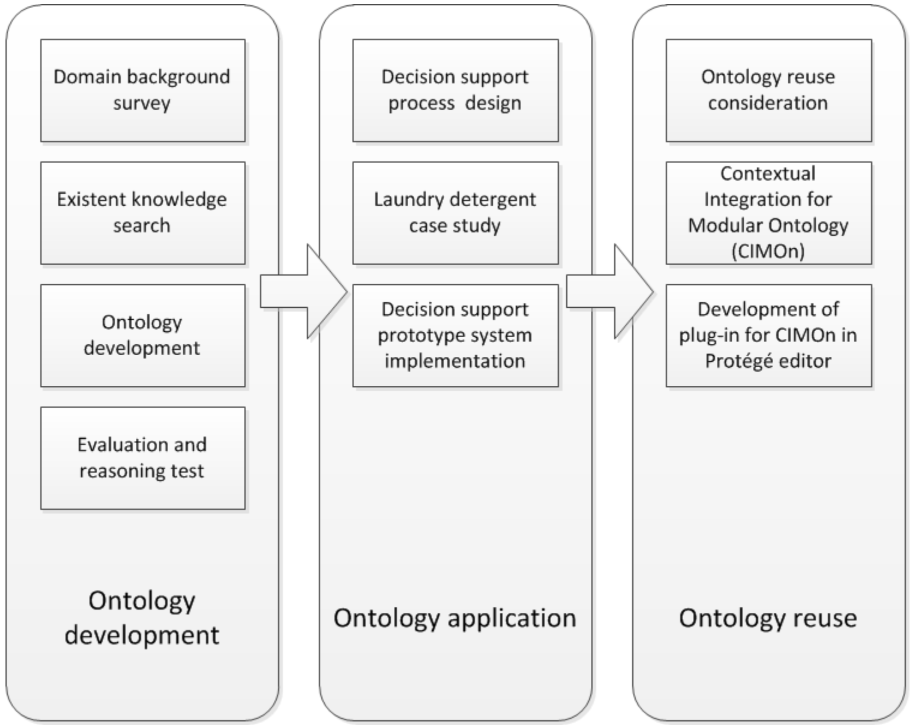


Figure 1: Mind map of our research methodology.

0.5 Contribution

The core contribution of our work is a decision support process based on modular ontology knowledge base. More precisely, instead of totally depending on human efforts in the eco-labeling process, we propose a decision support system as auxiliary tool to help experts make wiser decision in terms of the certification of eco-labels. The decision support system will be a powerful tool for the experts from EU Eco-labeling authorization, on the other hand, it can also be a simulation tool for those eco-label applicants. By generalizing the steps and tasks, the decision support process can also be applied to other product group.

To realize the functioning of this decision support system. A knowledge base composed of modular ontology has been constructed. Most of the domain knowledge about eco-labeling would be translated and stored in OWL ontologies. To better manage and reuse the modular ontology knowledge base, first, we have introduced a separation pattern between entity module and rule module by using owl:imports. This pattern actually makes a distinction between the descriptive & objective knowledge between subjective knowledge. It is proven to be in favor of ontology re-usability and extensibility. By using this separation pattern or scheme, we can build well-structured modularized ontologies. We also demonstrated how to do reasoning and inference in the criteria ontology by using Hermit reasoner. Since there is still a lack of ontology-based decision support application for eco-labeling, our work can provide some reference and experience for similar tasks. We have also published this ontology on GitHub, other researchers can access it and take modules they need for reuse.

We have proposed Contextual Integration for Modular Ontology (CIMOn) method to integrate and reuse different ontology modules. CIMOn is a method to integrate ontology modules by introducing an intermediating component called "context". By using "context", extra information that is needed only when ontologies integration happens can be independently stored. Thus, original ontology modules keep intact and can be reused for different purposes in different contexts at the same time. CIMOn is compatible with owl:imports, while a "filter" is introduced to achieve partly importing which means content of ontology can be partly integrated in context and this feature makes ontology reuse more flexible. Although there are quite several modular ontology and distributed ontology schemes or solutions, most of them stay in theoretical elaboration, few practical aspects are discussed and even fewer tools are available. In our work, besides CIMOn method, a tool as Protégé plug-in is developed. We consider publishing this tool after further improvement and more testing. To some extent, this is another important contribution of this work.

0.6 Thesis outline

In Chapter 1, we will have an overview of eco-labeling industry and its current status. The scope of eco-labeling is defined. Two main kinds of eco-labeling's concerns are discussed. Because of the large quantity of eco-labels, comprehensive and full inspection for each of them is impossible. In particular, EU Eco-label is chosen to be a representative case in our research. At the end of this chapter, main problematics and challenges of today's eco-labeling are identified.

The first two sections of Chapter 2 are about the state of art of decision support in terms of eco-labeling and ontology. Here we can find that even though decision support technology is already quite widely used, limited number of works and research about decision support systems applied in eco-labeling can be found. Most of the research on eco-labeling is about process improvement, green production, and social impact. However, general decision support system based on ontology has drawn considerable attention. In our research, we try to fill the gap so that ontology based decision support system can be used in eco-labeling and other similar labeling or certification schemes. In the third part of Chapter 2, we present a research overview about ontology and ontology engineering. Various aspects about ontology, its underlying logics and related technologies are presented. In our research, we care more about modular ontology or ontology modularization. Note that in the beginning of each subsequent chapter, there is also a state of art section. They will work as supplement and recall of Chapter 2.

Chapter 3 gives a detailed description about the ontology based knowledge base that we developed for the decision support system. A water-fall alike ontology engineering methodology was applied. We demonstrate how the knowledge about eco-labeling criteria was extracted and how the criteria ontology was built. Here we applied a modular approach to divide ontology into small pieces so that they can be reused more easily. We also demonstrate how to make use of reasoner to deduce new knowledge; how to judge whether a product is eco-labeling criteria compatible or not. At last, an evaluation and analysis of the knowledge base was conducted. We also discussed the added-value and expectation of such kind of knowledge base in an extended context.

Chapter 4 is actually an extend and in-depth dig of Chapter 3. In this chapter, we propose the CIMOn method trying to improve the efficiency and flexibility of ontology reuse and modularization. We will see that new construct called “context” is introduced as an intermediating environment to integrate ontology modules. An investigation of current modular and distributed ontology methodologies is also presented. We will discuss the features and deficiencies of these methods. Then we will explain in details how CIMOn works; what kinds of advantages it have; how to apply this scheme to ontology development and other engineering aspects.

In Chapter 5, we propose the decision support process based on the ontology knowledge base described in Chapter 3. Concretely, EU Eco-label laundry detergent product criteria is chosen as our case of study. A detailed description of the decision support process and the architecture of the system is presented. We have also shown how the criterion in the laundry detergent criteria document was translated into SWRL(Semantic Web Rule Language) rules and how to make use of reasoner and these rules to get the reasoning result as well as the arguments.

The content of Chapter 3, 4, and 5 actually reflect the main contribution of our work. In our research, we have exploited how to build modularized ontology out of criteria document. A entity-rule separation pattern was proposed; CIMOn method was proposed to address the re-usability related issues; Then, a complete decision support process is proposed to tackle the problematics and challenge that we have discussed in the first chapter.

Chapter 6 is about the prototype development and program implementation of our research work. There are two main implementations in our work. The first one is a plug-in used for the CIMOn’s context editing. The second one is a prototype implementation of the decision support process, i.e. decision support system prototype for EU Eco-labeled laundry detergent product. We will explain how we have realized them from design to coding as well as some engineering experience.

In the last chapter, we will do a brief review of this work, talk about its limits and unsolved problems. In the very ending part, we will identify several key tasks that could be interesting for other researchers in the future.

Chapter 1

Context & problematics of eco-labeling

1.1 Introduction

In a more and more industrialized world, ecology harmony is becoming an important aspect in production and diary life of modern society. Not only today's consumers expect the products to be of good quality and usability, but also they want them to be safe and environment-friendly. Since the first eco-label Blue Angel was awarded in Germany in 1978, many eco-labels covering various environmental aspects have been developed and put into service. To better manage and recognize Eco-labels being operated in different markets and countries, a Global Eco-labeling Network (GEN)[1] was established in 1994 as a worldwide non-profit interest group whose goal is to foster co-operation, information exchange and harmonization among members. Driven by the impel of governments and society organizations, the number of products and services certificated by eco-labels is also increasing rapidly.

Regardless of the catalog or classifications, each eco-labeling regime has two types of fundamental concerns: ecology harmony of the nature and economy enhancement of society. In this chapter, we will try to give an as comprehensive context review of eco-labeling as possible from various aspects. Problematics of eco-labeling will be identified and discussed. First, an official definition of eco-label is introduced. Application domains with regards to different labeling types are specified. Then, this chapter gives the current application status of various eco-labels and eco-labeling processes in service. Typically, EU Eco-label is chosen to be a representative case in our research. At last, a perspective of eco-labeling in the future and challenges is discussed.

[1] https://www.globalecolabelling.net/

1.2 What is eco-labeling?

According to Global Eco-labeling Network (GEN), "eco-labelling" is a voluntary method of environmental performance certification and labeling practiced around the world. An "eco-label" is a label which identifies overall, proven environmental preference of a product or service within a specific product/service category. There are different classifications of labels. In contrast to "green" symbols, or claim statements developed by manufacturers and service providers, the most credible labels are based on life cycle considerations; they are awarded by an impartial third-party in relation to certain products or services that are independently determined to meet transparent environmental leadership criteria [6]. The International Organization for Standardization (ISO) has identified three broad types of voluntary labels, with eco-labeling fitting under the Type I designation. The main scope of our research is also on TYPE I eco-labeling as TYPE I has the most labels in practice. The following enumerations are definition of Voluntary Environmental Performance Labeling from ISO. It is easy to see the difference between those three types.

- TYPE I: a voluntary, multiple-criteria based, third party program that awards a license that authorizes the use of environmental labels on products indicating overall environmental preferability of a product within a particular product category based on life cycle considerations.
- TYPE II: Informative environmental self-declaration claims.
- TYPE III: Voluntary programs that provide quantified environmental data of a product, under pre-set categories of parameters set by a qualified third party and based on life cycle assessment, and verified by that or another qualified third party [7].

Eco-labeling has numbers of benefits from various points of view. Firstly, eco-labeling is a good way to inform consumers of the environmental impacts of products. In the practice of some existent eco-labels, besides the environmental performance, the fitness of use and human health aspects are also included. All these information will help a consumer make decision in view of different willingness. Eco-labeling is generally cheaper than regulatory controls. By empowering customers and manufacturers to make environmentally supportive decisions, the need for regulation is kept to a minimum. This is beneficial to both government and industry [8]. Eco-labeling will also stimulate market development and encourage continuous improvement on product and service.

Now let's check some main concerns of eco-labels to better understand the essence of eco-labeling and its objective:

Human health and safety Before we consider the environmental and ecological issues, we should first make sure that a product or service is safe enough when it is about to be delivered to the market and finally to consumer. The definition of safety could include various aspects and could have subtle distinctions in different situation. For the final consumer, a product with eco-label usually guarantees that it neither contains nor diffuses substance that could hazard human health. Or, some product may indeed contain hazardous ingredient but the concentration is under certain dosage thread so that it shall be safe for an instructed usage.

Pollution and environmental performance Besides outstanding usage performance and good intention claiming that the product will not hazard human health, eco-labeled product and service usually guarantees a certificated environmental performance during the usage and afterwards. We can often see such eco-labels on various kinds of laundry detergents, paints, cosmetics, etc. Ecology environment, as human exploitation and reformation spreads rapidly over the planet, is becoming vulnerable and uncertain. To avoid ecology unbalance and its negative consequences which could threaten human survival, an eco-labeled certificated product usually claims that its ingredients and packaging are recyclable more or less. In some other cases, the waste of the products is claimed to be bio-degradable and non-toxic.

Energy consumption Some other eco-labels care more about decreasing the energy consumption and green house emission. Such eco-labeled products usually claim that they consume less energy in its production, delivery and usage stages than others. We can find such eco-labels on many electronic devices, especially household appliances.

Sustainability and efficiency Many today's eco-labels will consider the whole life-cycle of one product or service. This consideration is reasonable because modern society pays attention not only to product itself, but also the extraction of material, production process, till the final delivery and usage. This extended consideration requires not only the safety for the final consumers but also all the individuals who are possibly involved into the product's life-cycle. Recently, European Commission proposed an ambitious circular economy strategy [9] in which this life-cycle concern can be vividly demonstrated. The circular economy concept is a response to the aspiration for sustainable growth in the context of the growing pressure of production and consumption on the world's resources and environment.

More competent product or service Some eco-labels not only confine the environmental impact, but also guarantee the product's effectiveness in use. Those products certificated by such labels usually have the equivalent or better performance compared with the other products of the same kind. From the point of view of producer, here the labels somehow work like advertising. Though the application for such eco-labels are never for free, quite numbers enterprises show great interest in having their products or service eco-labeled to be more competent.

Market stimulation From a global point of view, eco-labeling could guide the market to develop to a more sustainable and efficient direction. As EU Eco-labeling has claimed, it targets the top 10 or 20 percent products in the market and the EU commission will regularly update the criteria threshold to keep such a percentage, which means eco-labels can be used to encourage innovation and improvement in respect of business and environmental performance.

At present, according to *Ecolabelindex*[2], there are over 450 eco-labels in 197 countries, and 25 industry sectors. In the report of Global Ecolabelling Network (GEN) in 2014 [10], certificated products worldwide exceeds 250,000. According to the statistics of "2010 Global Ecolabel Monitor" report [11] which was finished by World Resources Institute and Big Room Inc, most eco-labels are run by non-profit (58%) and for-profit (18%) organizations. 8% were government run. The majority (64%) were third-party certification systems. In Figure 1.1, we can tell that eco-labeling is more prosperous in developed markets.

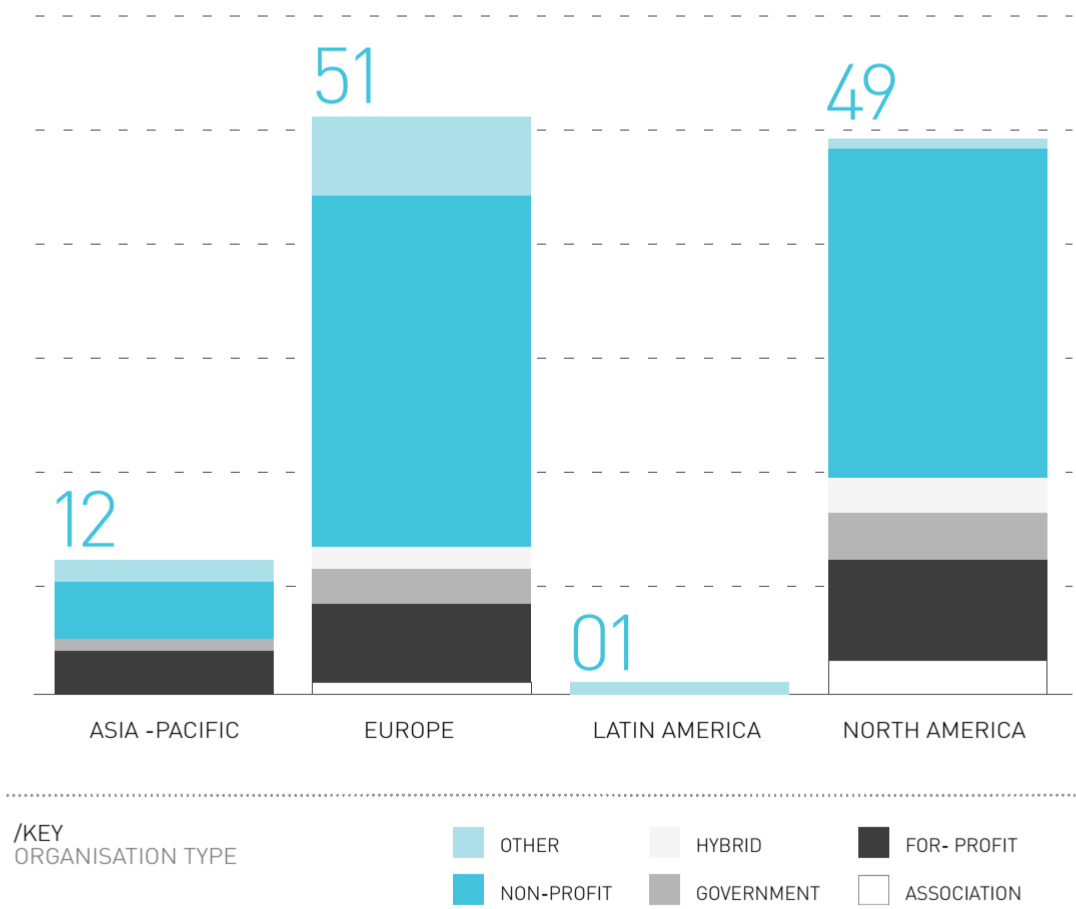


Figure 1.1: Number, type and location of organizations completing the global eco-label survey.

[2] http://www.ecolabelindex.com/ A global directory of ecolabels.

The 18th European Forum on Ecoinnovation [12], held in Barcelona 20-21 May 2015, was dedicated to the role of environmental labelling, management and information schemes in boosting innovation and competitiveness to help deliver a circular economy in Europe. The following outcomes of the Conference support the role of ISO14024 Type I ecolabels and the precepts of GEN [13]:

- Participants at the conference agreed that an environmental label must be credible (verified by a third party), and clear (easy to understand).
- Credibility would be further enhanced if the labels were used by public authorities in public procurement and if they covered a range of significant environmental impacts.
- There was a call for reducing fragmentation in environmental information and management schemes. The high number of labels and certification schemes generates confusion on the market
- There was a unanimous call in favor of tougher rules to tackle misleading green advertising and to develop minimum requirements that all labels would have to adhere to.
- There was widespread support for policymakers to prioritise high-impact products for labelling.
- B2B labels are as important as B2C labels also because it can be difficult to reflect all issues in a single label on an end product.
- Retailers play a pivotal role (they can both monitor producers and help consumers understand labels).
- In some cases, public commitments to sustainability by organizations could be an alternative to individual product labelling, but the same rules for credibility would apply. Many believe both product labelling and organizational commitments are necessary.
- Consumers are increasingly paying more attention to labels. Ecolabels are facing many other competing forces (use of VIPs for marketing, peer pressures, etc.) that can be turned to advantages if appropriately used (i.e. resulting in dedicated communication campaigns).

In the foregoing, we have introduced what is eco-label and on-going status of general eco-labeling. We have referred to different surveys and reports. A basic understanding of eco-labeling context is obtained.

1.3 EU Eco-label and labeling process

After a brief review of generic eco-label, we focus on the EU eco-label which relates to most of our research work. Since our research is not limited to a survey research, we must be more specific to a certain eco-label in order to propose concrete solutions. This section will introduce the background and details about EU Eco-label and the routine labeling process.

EU Eco-label is a successful example among all the eco-labels. Created in 1992, the EU Eco-label is the only official European ecological label authorized for use in every member country of the European Union [14]. The EU Eco-label covers a wide range of product groups, a full catalog of these product groups goes as following:

- **Personal care products:** Rinse-off cosmetic products; Absorbent hygiene products.
- **Cleaning up:** Hard surface cleaning products; Detergents for dishwashers; Industrial and institutional automatic dishwasher detergents; Hand dish-washing detergents; Laundry detergents; Industrial and institutional laundry detergents.
- **Clothing and textiles:** Textiles; Footwear.
- **Do-it-yourself:** Paints and varnishes.
- **Electronic equipment:** Imaging equipment; Personal, notebook and tablet computers; Televisions.
- **Coverings:** Wood-, cork- and bamboo-based floor coverings; Hard coverings.
- **Furniture and bed mattresses:** Furniture; Bed mattresses.
- **Gardening:** Growing media, Soil improvers and mulch.
- **Household appliances:** Heat pumps; Water-based heaters.
- **Lubricants:** Lubricants.
- **Other household items:** Sanitary tap-ware; Flushing toilets and urinals.
- **Paper products:** Converted paper; Newsprint paper; Printed paper; Copying and graphic paper; Tissue paper.
- **Holiday accommodation:** Tourist accommodation services.

The Commission of the European Communities launches the criteria revision procedure either itself or at the request of the European Union Eco-Labeling Board (EUEB), which comprises the certification bodies (Competent Bodies) in various European countries and the advisory forum (representing the stakeholders, i.e. SMEs, artisans and their professional organizations, trade unions, traders, retailers, importers, environmental protection groups, and consumer organizations). In each member country, the Competent Body is in charge of the concrete routine such as consulting, assessment, delivery and monitoring of each eco-label. EU Eco-labels are tools used to highlight a product's level of ecological quality. They guarantee a product's user quality, and also that it has a reduced environmental impact. Products are awarded eco-labels according to a multi-criteria approach (consumption of raw materials and energy, generation of waste, release into air and water) that factors in the product's entire life cycle (from the extraction of raw materials, through use and up to end-of-life).

Until 2011, there are over 1300 enterprises that have been issued EU Eco-label licenses. By September of 2014, there are already over 43,000 products or services being labeled [15]. France is always an important contributor to EU Eco-labeling. By March of 2016, 486 enterprises in France have obtained EU Eco-label licenses in various product groups and that makes France the first place as for the enterprises' possession of EU Eco-label licenses. However, compared to the enormous Europe market, the awarded eco-labels are still too few. We consider that qualified enterprises should be encouraged to obtain eco-labels to become more competitive. From global point of view, increasing the number of awarded products and enterprises should contribute to the reduction of environmental impacts.

As illustrated in Figure 1.2, the removal of certain product group (e.g. IPV:Indoor paints and varnishes, SSC: Soaps, shampoos, and hair conditioners, and OPV: Outdoor paints and varnishes.) which happened in 2016 indicates that the change or alteration of EU Eco-label criteria is continuous. It also implies that the change of knowledge and rules. Many other product groups keep increasing in the recent 4 years.

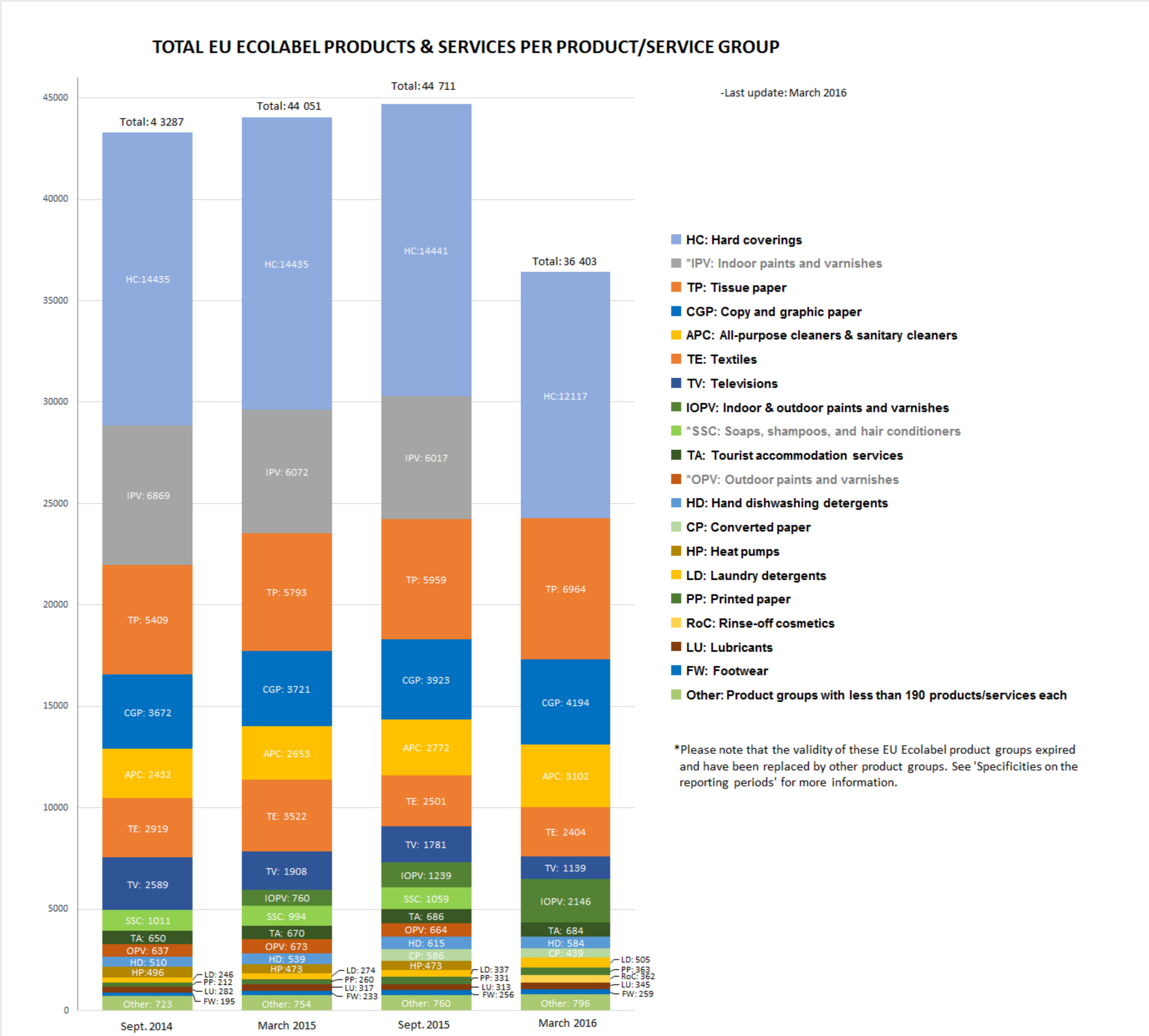


Figure 1.2: Total EU Eco-label products & services per product/service group

Throughout the product categories, the multi-criteria or guideline referred by EU Eco-label is usually stricter than the domain regulation. For example, the concentration limit of certain toxic chemicals as ingredients in eco-labeled detergent is lower by a magnitude than that is required in relevant standard regulation. In other words, one implicative effect via setting up eco-labeling is to keep only a small part of top ranked products and services able to be awarded. Such differences between EU Eco-label and standards consolidate its effect as a stimulation to the market and somehow a driving force to the producer.

The Commission mandates the EUEB to develop and regularly review eco-label criteria.

The Commission issues a call for tenders resulting in the selection of an advisory body, and a work group is formed. The advisory body conducts a feasibility study and then proposes fitness-for-use criteria and environmental criteria. Consultation continues throughout the drafting of the specifications, alongside the feasibility study and the development of criteria concurrent with regular feedback to the EUEB. On completing the work for a given product category, the Regulatory Committee summons representatives from every Member State and votes on whether to approve the guideline [16]. The guideline developed by the advisory body, together with the possible amendment or annex will be the baselines for our knowledge base. In other words, we will construct the ontologies with the concepts, relationships and rules extracted from these guidelines so as to make these experience and knowledge exploitable by the machine.

1.4 Challenges and better eco-labeling outlook

As we have seen previously, eco-labeling has been in practice for decades and there are many eco-labels of different types and many organizations who deliver these Eco-labels. While, according to some market surveys and statistics, certain eco-labels are not under proper management and operation. Problems and challenges exist. For instances, what if the awarded eco-label is misleading or cheating. Terms such as "recyclable", "biodegradable", and "ozone friendly" must be used accurately. When claims are used arbitrarily in advertising and labeling, customers will become confused, discouraged, and doubtful [17]. In the eyes of different stakeholders, the assessment methods and criteria of a product could be quite different. What techniques are the most appropriate and correct for certain product and service as the technology and even the product itself evolves fast? How to define a proper differentiation between standard and eco-labeling assessment criteria? All such issues require serious consideration to achieve better solutions in respect to various social aspects. Here in this section of this chapter, we will identify and discuss these issues.

Social propagation and interactive media Generally speaking, the certification of eco-labels are delivered to enterprises. A third-party organization or government department may take charge of the awarding and monitoring. While, for the consumers, they are often lack of detailed information about eco-labels and relevant labeling procedures and criteria. In addition, consumers are sometimes ignorant of eco-labels or they are totally confused by so many kinds of eco-labels. For example in France, according to a survey conducted by the French Environment and Energy Management Agency (ADEME) in 2012, only 52 percent of the population can give a relative precise definition of sustainable development, and only around 30 percent know the existence of EU Eco-labels [18]. Also, we should see that people from different ages have wide gap of awareness of eco-labels. For example, according to ADEME, the young people have a better understanding about eco-labels than the old ones. People with higher level education show much more interest on eco-label and environmental issues than those less educated. Therefore, there exists much room to improve and achieve a wider propagation about ecology, sustainable development and eco-labels in the population. On the other hand, according to the report "An overview of eco-labels and sustainability certifications in the global marketplace" [19], the eco-labelers are asked "what would help you to improve the overall effectiveness of your eco-labels program?" Public or consumer awareness wins 50 percentage as effectiveness improvement factors form the labeler's perspectives, which means a well population of eco-labels are also the request of eco-label issuers.

Almost every eco-label in the world has its own website on which consumers and enterprises would find description and publicity. But the problem is the consumers are often frightened by the overwhelming amount of information that few of them will take time to read and understand details about all kinds of guidelines and criteria. Thus when they see an eco-labeled product in the supermarket, they may still feel lost because the information is not efficiently propagated, or in other words, the information is not organized and presented in a proper way that the public could understand the meaning of the label and its advantages. [20] highlights the importance of not only the content, but also the accessibility of the information gathered in eco-labeling certification programs. Ultimately, the goal is not to increase the volume of information, but to help consumers make more informed decisions. [21] shows that the effectiveness of an eco-label depends both on how the information is presented and on the ability of the consumer to absorb and act upon it. There is also finding in [11] shows that over half of the eco-labels surveyed, including some prominent eco-labels, were unreachable, difficult to reach, or uncooperative when asked about core metrics. In and of itself this indicates the need for improvement in transparency and accountability across the voluntary standards sector.

The other side of the market, the enterprises, how many of them show willingness to be certificated? For particular enterprise, which product family would have possibility to be certificated? For the organizations or government departments who play the role of eco-labelers, based on what criteria should they set up a new eco-label category? To answer these questions, the enterprise had better to thoroughly understand the market and the consumer. But how do they get access to such data? The traditional propagation method will make use of public media, while, they usually put too much efforts on "push" which means they propagate the information like throwing a stone to the sea, regardless of the echoes and ripples. What if we provide a platform or system for both consumers and enterprises as well as those labeling organizations on which each participant could share their experience, edit their knowledge and post their query for certain information. All these data and information should be traced, stored and analyzed so that related stakeholders can access. Via the supplement of such an interactive propagation platform, we believe that we can achieve a better population of eco-labeling so as to the sustainable development and ecology harmony.

Green consumption For the conscious consumers, environmental performance should be a necessary but not sufficient condition for purchase. [22] notes that consumers purchase functional products for functional reasons. This means that a laundry detergent that is 100% biodegradable and manufactured with a minuscule carbon footprint will never outcompete any other product if it is not an effective detergent. A product that cannot deliver consumers' needs will fail in the marketplace, no matter how eco-friendly it is. How to encourage green consumption and populate eco-labeled products among consumers? A very common question asked by most consumers is if the eco-labeled products have a higher price or not. Another interesting result shown in the report of ADEME is when the unemployment rate is high so that the purchase power is more feeble, conscious consumers don't intend to consider much about environment-friendly things. This seem to help explain why eco-labels are more popular in developed markets, e.g. North America and Europe. Then the question is if eco-labeled products' price is relative higher than non-labeled ones, what would happen when people are confronted with negative economy situation? How to hold the market share of labeled products even in face of economic crisis?

To hold the ground of green consumption, we should extend the scale of eco-labeled products and services. This also requires a higher requirement of eco-labeling efficiency. Via a platform that owns a large number of users especially the consumers, the labelers and enterprises could get timely feedback and more importantly they can prepare for innovation and necessary improvement to achieve a better trade-off between economy and environment.

Internal organization, evaluation and evolution How to spread best practice to other countries or markets? How to assess the overall performance of eco-labeling from various aspects, which means an evaluation of the eco-labeling evaluation itself?

The number of eco-labels in the market keeps increasing and the eco-labeling itself is becoming a business. The rapid increase in proliferation of eco-labels in the market is accompanied by a high level of redundancy among eco-labels. Twenty-eight percent of responding eco-labels recognized other labels as being equivalent, while 33% of responding labels were recognized by other labels as equivalent. Redundancy exists means that there could be competition between equivalent eco-labels. How to stand on the high ground is a potential issue in the future [19].

Also in [19], some interesting numbers and statistics can be found. There is a fair amount of variation in the length of time it takes for a manufacturer to become certified. Among single-standard eco-labels, the most common response among labels for time required to certification was three to six months, with 37% of respondents falling into this category (see Figure 1.3). However, 12% of labels offer certification in less than two weeks, with some providing next-day certification. At the other end of the spectrum, some labels require one to two years for certification. Although the average time to certification across single-standard labels is 4.33 months, the standard deviation is 4.37 months, indicating that there is still a significant lack of uniformity in the market. If we apply a decision support tool as an aid for the labeling process, we believe the response time will be dramatically shortened. This will be especially helpful when we apply eco-labeling or shift best practice in a fast developing market i.e. China and India.

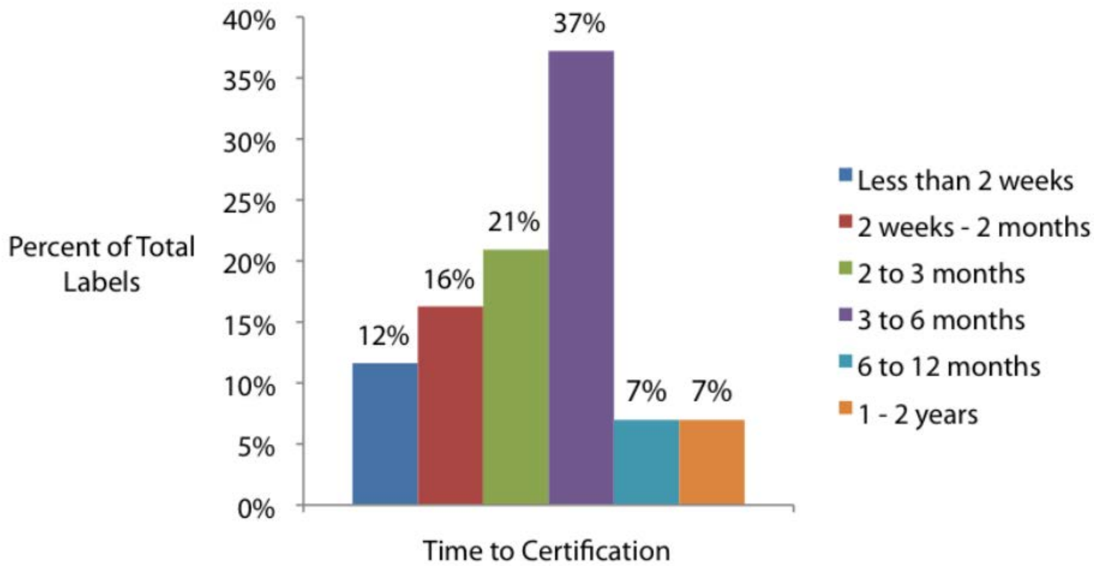


Figure 1.3: Average time to eco-labeling certification

One criterion identified for a successful label is the extent to which the organization can demonstrate positive on-the-ground impacts resulting from its labeling program. It was expected that the majority of labels would have conducted studies to assess the benefits of their labeling programs. Instead, the survey data analysis indicates that only 44% of single-standard labels have conducted an impact study. Fifty-five percent of responding labels indicated that they had not conducted such a study; 22% indicated that they had plans to do so. One-third of labelers surveyed had made no attempt to monitor or evaluate the environmental and social benefits of eco-labels programs and have no intention of doing so [19]. Here we can see that there is still much room to improve eco-labeling in terms of rapidity and quality.

Collaboration and alignment Although we have seen many eco-labels and certification based on different standards. Almost no collaboration or interoperability exists among them. Every labeling system seems to be independent from each other. According to [11], less than 30% of eco-labels recognize or are recognized by other labeling organizations, indicating an opportunity for increased collaboration amongst eco-labels to reduce confusion amongst users of the eco-labels. With nearly all eco-labeling organizations requiring some form of metrics reporting, there is further opportunity to collaborate and create more aligned standards. Here is another statement made by ISEAL [23]: " Standards organizations are usually small to medium in size and based on a not for profit business model which often places constraints on time and resources. Aligning themselves and collaborating with other similar organizations and standards to avoid unnecessary duplication of work, to share rather than compete, and to increase harmonization of work can result in improved efficiency for the sustainability standards community as a whole."

Cost of certification “The real cost of certification is not so much in the certification process itself as it is the preparation for the certification, in preparing the data and information required for the audit. Thus capacity building to assist producers, especially SMEs, in being better prepared to make data readily available for the audits can greatly reduce the time taken for audit which is directly proportional to the cost of certification.” stated by ISEAL [23] again. According to our survey, EU Eco-label is already cost-effective and even more economic for SME (Figure 1.4), while people will never refuse a cheaper price.

PRODUCTS

Application/Renewal Fee			Extension/Modification Fee			Annual Fee			Inspection Fee	
Standard	SMEs and Operators from Developing Countries	Micro-Enterprises	Standard	SMEs and Operators from Developing Countries	Micro-Enterprises	Standard	SMEs and Operators from Developing Countries	Micro-Enterprises	Europe	Outside Europe
1 200,00	600,00	350,00	910,00	455,00	305,00	1 515,00	760,00	355,00	Depending on the application 1 380,00/day + travel costs	

TOURIST ACCOMMODATION SERVICES AND CAMPSITE SERVICES

Application/Renewal Fee			Extension/Modification Fee			Annual Fee			Inspection Fee	
Standard	SMEs and Operators from Developing Countries	Micro-Enterprises	Standard	SMEs and Operators from Developing Countries	Micro-Enterprises	Standard	SMEs and Operators from Developing Countries	Micro-Enterprises	Europe	Outside Europe
1 200,00	600,00	200,00	-	-	-	1 500,00	500,00 – 750,00 (according to the size)	200,00	639,00 - 1 595,00 (according to the size) + travel costs Costs outside Europe still under study	

Currency: EUR.

Figure 1.4: EU Eco-label fee in France

1.5 Conclusion

In this chapter, we have introduced the context and domain of our research from a practical point of view: eco-label, representative EU Eco-label and their context. We have identified the main problematics and challenge of eco-labels. They can more or less reflect the trends of eco-label development in the future. According to our survey, in general, eco-labels are getting more popular. In order to let more and more people choose eco-labeled products and then achieve an ecological friendly economy, a more efficient propagation is needed. It is also found that collaboration and alignment of eco-labels will be valuable. In particular, a reduction of certification cost and acceleration of eco-labeling process can stimulate the population and democratization of eco-labels. To our knowledge, this could be the key to all the other challenges. In the next chapter, we will present a survey on how to reduce certification cost and accelerate eco-labeling process via decision support approach. How to build decision support system by means of ontology and relative technologies. More importantly, scientific issues related to ontology and ontology engineering will be identified and revealed.

Chapter 2

State of the Art

2.1 Introduction

In this chapter, we will introduce the problematics and context of our research from a scientific point of view. Some background and basics will be presented here. In the beginning of each next chapter, further more specific discussion and our comments will be presented as complement. We will start from decision support system for eco-labeling and decision support with ontology. Then a current status of ontology, and modular ontology will be introduced. At the end of this state of art survey, the main scientific problematics and issues will be identified.

2.2 Decision support for eco-labeling

Although a lot of efforts has been done in the field of decision support, to our knowledge, few concentrate on eco-labeling.

In [24], authors present and discuss the implementation of product life cycle assessment for eco-labeling using DES (Discrete Event Simulation). CMSD (Core Manufacturing Simulation Data) [25] is used to facilitate the definition of manufacturing information related to production operations in order to address interoperability issues between simulation and other manufacturing applications. But this work is dedicated to type III eco-labels exclusively.

A decision support tool called EDSS is available online[1]. EDSS is a tool that intends to help companies or organizations choose the most appropriate eco-label, among the hundreds available internationally, either for specific products or for the business as a whole entity.

[1] http://1-dot-bright-eon-851.appspot.com/

It is a very interesting tool, because as we have seen in the previous chapter, there are too many eco-labels available in the world. Unfortunately, this tool does not focus on product's evaluation and certification, instead, it's aiming at eco-labels themselves. Also, the user of EDSS is limited to companies or organizations.

In the work of [26], a tool to verify the compliance of a product with given norms and standards is described. It is shown how the knowledge contained in eco-labeling standards and norms in textual form can be translated into constraints. NIAM/ORM (Object-role Modeling) can be used for formalizing the product data and the verification of the eco-label compliance turns into a CSP (Constraint Satisfaction Problem). In their later work in [27], the authors present how CLAIRE (Combining Logical Assertions, Inheritance, Relations and Entities) language is used to solve the CSP. The tool presented in their work can be regarded as some kind of decision support tool. While, the underlying mechanism is not related to ontology.

In [28], the paper presents the development of certification and inspection ontology to support smart disclosure of product information. The primary focus and contribution of this research is in systematically describing the ontological modeling process of certification and inspection schemes to support smart disclosure of product information that is valued by a consumer as she makes a purchasing decision, which is very alike to the purpose of eco-labeling. Although they did not develop a complete decision support process or system, reasoner has been applied upon the ontology and complex DL query can be used to access the knowledge base. And that actually makes it possible to further develop decision support function based on the knowledge base.

[29] describes an environmental knowledge management tool capable of providing planners and production managers the knowledge related to the potential environmental impact of the manufacturing choices in a distributed manufacturing scenario. Although it is not directly related to eco-labeling, they both share a similar environmental performance assessment objective. The authors point out that the length of assessment for complex product and associated process chains could be very long and there is a lack of environmental impact assessment function for available PLM and ERP system. Ontology based knowledge-base is developed to store knowledge generated from or through interaction with the LCA (Life Cycle Assessment) simulation tools. Ontologies are enriched with semantic rules for product and process classification and mapping to the environmental concepts, to infer simulation models for the LCA tools.

In general, very few systemic approaches or solutions have been described and researched according to our search. We conclude that there is still big gap between decision support and eco-labeling, and our research is dedicated to try to fill such gap.

2.3 Decision support with ontology

Actually, decision support with ontology can be seen as a sub-domain of ontology application which is a very practical field carrying lots of engineering characteristics. It is quite similar to expert system, while some difference exists. Although rule based technology contributed to the initial success of expert systems, people noticed the difficulty in the maintenance of a rule base and that in sharing and reusing the knowledge in the knowledge base so that all the knowledge bases had to be built from scratch. Knowledge engineering has started to evolve from rule base technology to knowledge modeling since then to overcome these difficulties [30]. The Semantic Web shares many goals with Decision Support Systems (DSS), e.g., being able to precisely interpret information, in order to deliver relevant, reliable and accurate information to a user when and where it is needed. DSS have in addition more specific goals, as the information needed is targeted to making a particular decision. Semantic Web technologies have been used in DSS during the past decade to solve a number of different tasks, such as information integration and sharing, web service annotation and discovery, and knowledge representation and reasoning [31]. According to our experience, an advantage of decision support with ontology is the re-usability and interoperability of knowledge. That is also a significant advantage of ontology itself. If an ontology is very use-dependent or task dependent, it will be very hard to be reused. While, as we will see in the following part of this chapter, modular and distributed ontology as well as related engineering methodologies have been invented. When ontology knowledge base is no more limited to specific application tasks, reuse and composition of knowledge become easier.

According to our search and survey, ontology and semantic technologies have been used in broad range of domains. Although it is not explicitly declared, many of these practice and applications have implemented or they can be used as decision support in various forms. In [32], OWL/SWRL[2] modelling paradigm representing EDF MUDU (French acronym for user data unified model) is developed. This ontology model is made of a set of coherent user dictionaries that are used to generate business catalogues for each nuclear power plant project and each business activity. These dictionaries are constrained and designed following a meta-model which is composed of physical or abstract objects (components, relations and connection points), attributes (characteristics, graphical representations, and connection types), and rules. The authors tried to use SWRL rules to express complex business rules and applied the execution of SWRL rules in an industrial use case in the nuclear industry. Since SWRL provides association rules, and allows to associate instances to new classes and to create properties between instances, rule based decision support was realized implicitly. In this research, a literature review of SWRL as a rule language is made, some limitation of SWRL are discussed. While, there is little discussion and demonstration about the explanation and argumentation for the rules.

In [33], Ontology Driven Software Engineering (ODSE) is described. According to the authors, developed software components are annotated with semantic information from ontology. Appropriate software components are discovered based on semantic description of

[2]https://www.w3.org/Submission/SWRL/

them. Ontology can be a basis for verification (e.g. consistency checking) of system requirements or/and design. In comparison to the traditional way of software development, this approach partially automates development processes, facilitates reuse of components through metadata-based discovery, and raises the level of abstraction of smart application development. All these aim at making development easier and faster, also at enabling development by non-programmers. We have noticed that the software component search and system requirement verification can be actually seen as some decision making process.

[34] presents a Decision Support Ontology (DSO) which is developed to facilitate decision making within collaborative design. The DSO includes decision-related information such as the design issue, alternatives, evaluation, criteria and preferences. It also includes decision rationale and assumptions, as well as any constraints created by the decision and the decision outcome. In this research the DSO is structured to support the documentation of information independent of the decision method. The DSO can be extended through a supplementary ontology to capture information specific to a particular decision method. DSO is more like a model to store the decision and its various aspects, while, how to make the decision and how to make use of these decision related information is not elaborated in details.

[35] reports the development of a decision support ontology developed in web ontology language OWL-DL (Description Logic). The ontology is combined within a preoperative risk assessment software system with a DL reasoner in order to provide a number of clinical decision support functionalities, including risk assessment, recommended tests and recommended clinical precaution protocols. In the system, decision support is usually provided in a 2 step process. The first step typically calculates risk scores or derives risk grades using numerical formulas. At this stage, the system does not use the decision support ontology but merely computes values using an open source Java-based rule engine (JBoss). Once the risk grades and categories have been derived from the first risk calculation step, the system then performs decision support using the open-source Java-based PELLET reasoner to reason on the decision support ontology given a patient OWL medical history profile. According to the authors, numeric risk score calculation is currently easily done by the system using an open source Java-based rule engine, JBoss Rules, other tasks including categorization, classification and logical inference were beyond the capacity of the system prior to introduction of semantic technology. Also, there are many reasons why using an ontology for the decision support part of the system is appealing. They include: (i) the decision support ontology is a conceptually appealing formalism which clinicians can easily relate to, (ii) it helps to break down the complexity of managing a large number of guidelines and rules by organizing them in well-structured hierarchies and taxonomies (iii) it enables efficient reuse of information, etc. It can be found from this research that, reasoning and semantic inference is critical to the implementation of decision support functionality.

There is a very interesting survey work in [31]. The authors found that, when studying the contribution of Semantic Web technologies to the DSS field, it is noted that two high-impact areas have been *ontologies and semantics* and *Semantic Web data*, with particular emphasis on the former. Many DSS applications use ontologies and rules as a means for making the DSS “intelligent” in some data analytics sense. While many Semantic Web applications use rather light-weight solutions and ontologies, DSS on the other hand have often been used in more closed scenarios than the Web and have many times utilized quite complex ontological reasoning and rule bases, more similar to Expert Systems of AI than today's Semantic Web applications. The survey has also indicated that most DSS today seem not to have an "open world view" but seem rather static, where the sources are determined at design time and data is not usually linked or browsable, e.g., through some kind of drill-down with increased levels of detail in data, or through associative relations. Also, explanations of derived information, and drill-down in terms of exploring the underlying data or its sources, seems uncommon. It implies that further explanation for the decision making and open access for the decision related data is lacking. System must be able to explain how something was derived and what the user should potentially do about the situation. It is also indicated that, one of the most important basic needs of DSS, which cuts across all application domains of the interviewees, is the need for easier and more flexible information integration methods. Data interoperability and re-usability are also very important.

2.4 Ontology and Modular ontology

Since the new century, Ontology and Semantic Web have become hot topics in both academic and industry. We believe that Semantic Web could be a promising approach to realize more smart application or even stronger artificial intelligence. Ontology, with its origin in Semantic Web and characteristics, is considered to be an ideal scheme of knowledge representation. In this section, recent progress of ontology, ontology engineering and some other relative topics are presented.

2.4.1 Semantic Web

The Semantic Web, according to Tim Berners-Lee, is a web of data, in some way like a global database. The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users [36]. In addition to the classic "Web of documents" W3C is helping to build a technology stack to support a "Web of data", the sort of data we find in databases. The ultimate goal of the Web of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network. The term "Semantic Web" refers to W3C's vision of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as RDF, SPARQL, OWL, and SKOS [37].

As stated on the web site of W3C[3]: "Linked Data lies at the heart of what Semantic Web is all about: large scale integration of, and reasoning on, data on the Web." "To make the

[3]https://www.w3.org/standards/semanticweb/data

Web of Data a reality, it is important to have the huge amount of data on the Web available in a standard format, reachable and manageable by Semantic Web tools. Furthermore, not only does the Semantic Web need access to data, but relationships among data should be made available, too, to create a Web of Data (as opposed to a sheer collection of datasets). This collection of interrelated datasets on the Web can also be referred to as Linked Data."

In fact, similar pattern of Linked Data is already out there for human agent since Internet was created. As for most of today's web applications, web pages usually contains large amount of hyper-links. Hyper-links or URLs work as anonymous relations to make pieces of information or data related. While, the meaning upon such relationships have to be parsed and comprehended by human. To better understand Semantic Web and Linked Data, we take an example: When we request an item in Wikipedia, the search engine will acquire precisely the web pages we want. (Search engine technology is indeed a great innovation, and it shall act as some foundation of Semantic Web.) The human user read the content and get the ideas on the pages. Usually, the user encounter some items that he doesn't know, which means the information on this page is not enough to accomplish our goal. Very often, he'll will click the hyper-links on current page, jump to another relative page and may as well repeat this action multiple times until he feels that all those information he has browsed is sufficient to memorize or to understand the item he searched at first. The description above is a typical web surfing behavior that happens every day. While, the point of introducing Semantic Web is, what if it's not a human that is browsing the web? How we drive a machine instead of human brain to acquire information on web?

One of the advantage of Semantic or Linked Data is expandability that the knowledge scale and data volume is almost unlimited as data has rich relationships with other repositories and the whole Semantic Web can be regarded as potential data source. This also means that an autonomous information process and exploitation is possible. While exploiting the Semantic Web, an agent can easily get reach to other structured information that may interest him via certain form of relations. The difference between hyper-links in plain text content on a web page and relations referred in Semantic Web is that the relationship in Semantic Web can carry a predefined structure and semantic meaning readable by machine. A hyper-link can only convey the fact that two concepts are related, but how? "Son" and "father" are related, but what exactly the relationship is? In Semantic Web, every piece of information, every concept and relationship can be defined and semantic profile is accessible to both human and machine are given.

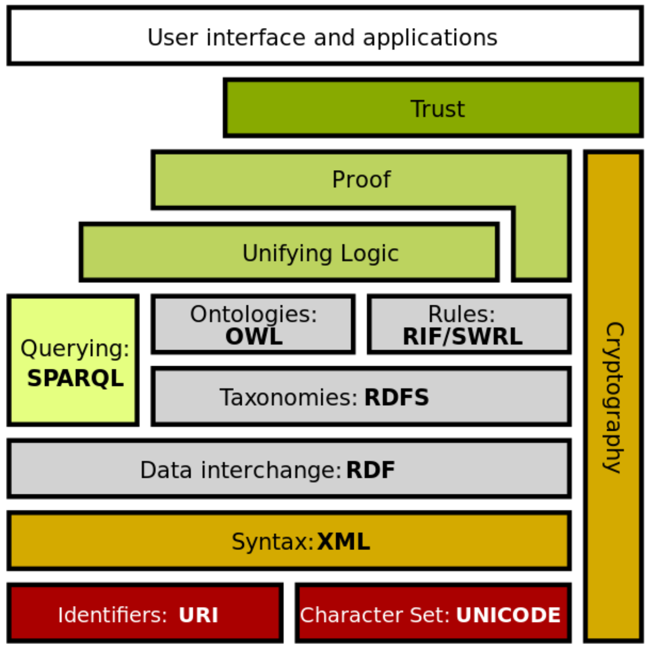


Figure 2.1: Tim Berners-Lee's Semantic Web Stack, or Semantic Web Layer Cake

It has been widely accepted that the architecture of the Semantic Web will be based on a hierarchy of languages, each of which exploits both the features and extends the capabilities of the layers below. This has been illustrated in Tim Berners-Lee's famous "Semantic Web Stack" in Figure 2.1. It shows how technologies that are standardized for Semantic Web are organized to make the Semantic Web possible. It also shows how Semantic Web is an extension (not replacement) of classical hypertext web. The stack is still evolving as the layers are concretized [38]. So it will not be strange to see slight changes and alteration in another stack illustration. The technologies from the bottom of the stack up to OWL are currently being standardized and being used to build Semantic Web applications. It is still not clear how the top of the stack is going to be implemented. All layers of the stack need to be implemented to achieve full visions of the Semantic Web. As a result of the work of the W3C Web Ontology Working Group, the "Ontology" layer has now been instantiated with the Web Ontology Language OWL. Since then, attention has turned to the rules layer, and much effort has been devoted to the design of suitable rules languages.

In this Semantic Web Stack architecture, we can see that ontology only hold part of the technologies, tools and languages that support the Semantic Web. Our research work in this thesis mainly focuses on ontology and its application. Before we dive into the domain of ontology, we think it is necessary to understand where our research is positioned in a broader Semantic Web background. In the following part of this section, we will have a brief but not complete review of ontology related theories, technologies and tools.

2.4.2 Ontology

Traditional database based on relationship model has been invented and applied in business for decades. Generally speaking, relational database is quite successful in terms of modeling and querying efficiency. However, as data keeps increasing in an explosive way, relational database's drawbacks become apparent. It is found that relational database is not flexible and especially clumsy in data exchange and inference aspects. With the rapid development of Semantic Web and Linked Data, the interoperability, reusability and modularity of knowledge are becoming more and more important. Thus, industry is calling for new representation scheme or repository solution for data and knowledge.

In fact, the term of ontology is not a newly created word. We have same term for " philosophical study of the nature of being, becoming, existence and/or reality, as well as the basic categories of being and their relations"[4]. The actual definition of ontology in term of knowledge representation in computer science was even given before the thrive of Semantic Web. In Computer Science, we refer to an ontology as a special kind of information object or computational artifact [39]. Back early in 1998, Studer et al. [5] had the definition stating that: "An ontology is a formal, explicit specification of a shared conceptualization." For the notion of a conceptualization according to Genesereth and Nilsson [40], who claim: "A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose." According to the author, ontology is a knowledge base that organize human conceptualization in a structured way. In [39], authors give a generic definition of ontology:

Let C be a conceptualization, and L a logical language with vocabulary V and ontological commitment K. An ontology OK for C with vocabulary V and ontological commitment K is a logical theory consisting of a set of formulas of L, designed so that the set of its models approximates as well as possible the set of intended models of L according to K.

Ontology's syntactic characteristics enable it to build knowledge base in a more natural way which is accessible to both human users and machines. Sharing common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies [41]. Hard-coding assumptions about the world in programming-language code makes these assumptions not only hard to find and understand but also hard to change, in particular for someone without programming expertise. Often an ontology of the domain is not a goal in itself. Developing an ontology is akin to defining a set of data and their structure for other programs to use [42].

The Semantic Web envisions a world wide distributed architecture where data and computational resources will easily inter-operate based on semantic marking up of web resources using ontologies. Ontologies are a formalization of the semantics of application domains (e.g., tourism, biology, medicine) through the definition of classes and relations modeling

[4] https://www.merriam-webster.com/dictionary/ontology

the domain objects and properties that are considered as meaningful for the application [43]. According to our understanding, ontology is like a modeling language standard which specifies the meta-information and characteristics of different knowledge representation. Generally speaking, an important objective of ontology is to make certain conceptualization identifiable and accessible by machines. We can use an analogy saying that an encyclopedia is a shared conceptualization about a domain comprehensible to human and an ontology shall be a shared conceptualization recognizable to application programs. In human society, the same knowledge can be filed in multiple languages. When an ontology is developed, it can be also edited in different languages or syntaxes.

In [44], a classification (Figure 2.2) based on the scope of the objects described by ontology was presented. For instance, the scope of a local ontology is narrower than the scope of a domain ontology; domain ontologies have more specific concepts than core reference ontologies, which contains the fundamental concept of a domain. Foundational ontologies can be viewed as meta ontologies that describe the top level concepts or primitives used to define other ontologies. Finally, general ontologies are not dedicated to a specific domain thus its concepts can be as general as those of core reference ontologies.

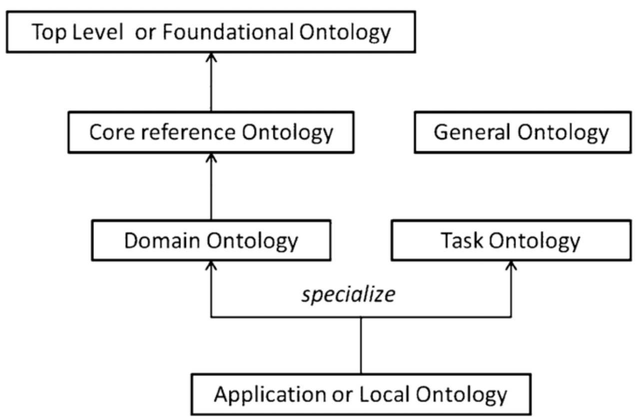


Figure 2.2: Ontology classification based on domain scope

In the last decades, so many ontologies and knowledge repositories have been developed and studied. Simultaneously, much problems are encountered when knowledge engineers as well as general users want to understand and employ the ontologies into their own software development. One reason of such difficulties is the semantic confusion among domains. Another reason, according to the author, is that there is still lack of a comprehensive and widely accepted standard system for ontology construction, e.g. ontology languages are developed based on logics having different expressiveness, which somehow block the compatibility for data exchange and reasoning. One of the thorniest problems is how we build ontology and utilize it to furthest maintain a well reusability. Due to the initial nature of being shared, certain formation of knowledge shall be meaningless if it could not be exploited and reused. Such a vision leads to a lot of discussion about ontology itself and the methodologies by which we can build correct and efficient ontologies. We will discuss such issues in ontology engineering section.

2.4.3 Resource Description Framework (RDF)

As we can see from Semantic Web stack, at the bottom layer of the stack lay components of URI, UNICODE, and XML. These elements are fundamental building blocks supporting today's web architecture as well. So the Semantic Web is not an independent artifact, instead, it shall be an extension of the web today. RDF (Resource Description Framework) is a hinge component for further development of other components laying in higher level of the stack. The Resource Description Framework (RDF) is a family of Worldwide Web Consortium (W3C) specifications originally designed as metadata data model. It has come to be used as a general method for conceptual description or modeling of information which is implemented in web resources, using a variety of syntax notations and data serialization formats. RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ. It specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. RDF extends the linking structure of the Web by using URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a "triple"). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications. This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations. The W3C published a specification of RDF's data model and an XML serialization as a recommendation in 1999 [45]. By the time this thesis is written, latest version of RDF is 1.1 (RDF Schema 1.1) [46], and it has evolved to be a rich extension of basic RDF vocabulary.

According to W3C Working Group Note [47], the items below illustrates different uses of RDF, aimed at different communities of practice (the last item is very important in our work):

- Adding machine-readable information to Web pages using, for example, the popular *schema.org*[5] vocabulary, enabling them to be displayed in an enhanced format on search engines or to be processed automatically by third-party applications.
- Enriching a dataset by linking it to third-party datasets. For example, a dataset about paintings could be enriched by linking them to the corresponding artists in Wikidata, therefore giving access to a wide range of information about them and related resources.
- Interlinking API feeds, making sure that clients can easily discover how to access more information.
- Using the datasets currently published as Linked Data, for example building aggregations of data around specific topics.
- Building distributed social networks by interlinking RDF descriptions of people across multiple Web sites.

[5]http://schema.org/docs/schemas.html

- Providing a standards-compliant way for exchanging data between databases.
- Interlinking various datasets within an organization, enabling cross-dataset queries to be performed using SPARQL. (SPARQL Protocol and RDF Query Language is an RDF query language, able to retrieve and manipulate data stored in Resource Description Framework (RDF) format.)

2.4.4 Ontology Web Language (OWL)

In Semantic Web Stack, ontologies and rules lay above RDF component. To a certain extent, it's RDF substances that constitute the concrete body of the semantic web. Based on the RDF's graph-like structure and with the help of SPARQL, we can access data in quite flexible manner. While, to attain more powerful functions as described in Semantic Web, ontology only in RDF is not enough, instead, a proper conceptualization of RDF substances is necessary. Ontologies are formalized vocabularies of terms, often covering a specific domain and shared by a community of users. They specify the definitions of terms by describing their relationship with other terms in the ontology. OWL 2 is an extension and revision of the OWL Web Ontology Language developed by the W3C Web Ontology Working Group and published in 2004. OWL (Ontology Web Language) is part of the W3C's Semantic Web technology stack, which includes RDF and SPARQL. Figure 2.3 gives an overview of the OWL 2 language, showing its main building blocks and how they relate to each other. The ellipse in the center represents the abstract notion of an ontology, which can be thought of either as an abstract structure or as an RDF graph. At the top, there are various concrete syntaxes that can be used to serialize and exchange ontologies. At the bottom, there are the two semantic specifications that define the meaning of OWL 2 ontologies. Any OWL 2 ontology can also be viewed as an RDF graph. The RDF-Based Semantics can be applied to any OWL 2 Ontology, without restrictions, as any OWL 2 Ontology can be mapped to RDF. OWL 2 follows open-world assumption which means if some fact is not present in a database, it is usually considered false (the so-called closed-world assumption) whereas in the case of an OWL 2 document it may simply be missing (but possibly true) [48].

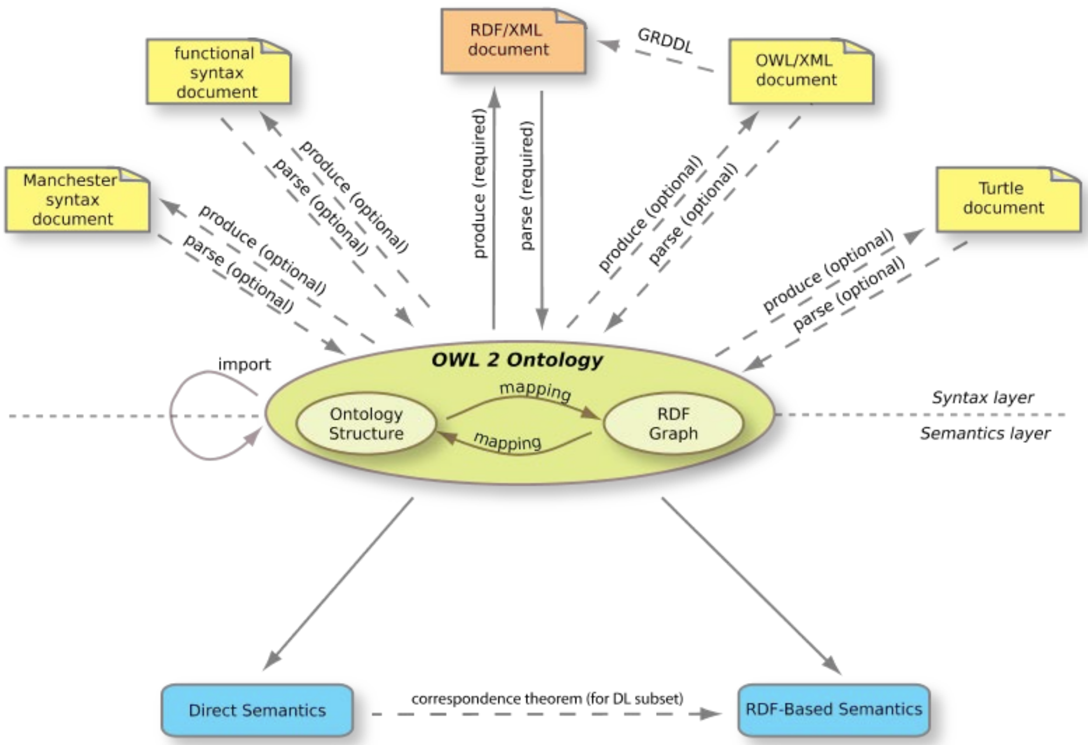


Figure 2.3: The Structure of OWL 2

According to the W3C work group declaiming in OWL 2 Web Ontology Language Document Overview (Second Edition) [49]:

Syntax layer: In practice, a concrete syntax is needed in order to store OWL 2 ontologies and to exchange them among tools and applications. The primary exchange syntax for OWL 2 is RDF/XML; this is indeed the only syntax that must be supported by all OWL 2. While RDF/XML provides for interoperability among OWL 2 tools, other concrete syntaxes may also be used. These include alternative RDF serializations, such as Turtle; an XML serialization; and a more "readable" syntax, called the Manchester Syntax, which is used in several ontology editing tools. Finally, the functional-style syntax can also be used for serialization, although its main purpose is specifying the structure of the structure of the language.

Table 2.1: Different syntaxes of OWL 2

Name of Syntax	Specification	Status	Purpose
RDF/XML	Mapping to RDF Graphs, RDF/XML	Mandatory	Interchange (can be written and read by all conformant OWL 2 software)
OWL/XML	XML Serialization	Optional	Easier to process using XML tools
Functional Syntax	Structural Specification	Optional	Easier to see the formal structure of ontologies
Manchester Syntax	Manchester Syntax	Optional	Easier to read/write DL Ontologies
Turtle	Mapping to RDF Graphs, Turtle	Optional, Not from OWL-WG	Easier to read/write RDF triples

Semantics layer: Direct Semantics and the RDF-Based Semantics provide two alternative ways of assigning meaning to OWL 2 ontologies, with a correspondence theorem providing a link between the two. These two semantics are used by reasoners and other tools, e.g., to answer class consistency, subsumption and instance retrieval queries.

The Direct Semantics assigns meaning directly to ontology structures, resulting in a semantics compatible with the model theoretic semantics of the $\mathcal{SROIQ}$ description logic—a fragment of first order logic with useful computational properties. The advantage of this close connection is that the extensive description logic literature and implementation experience can be directly exploited by OWL 2 tools. However, some conditions must be placed on ontology structures in order to ensure that they can be translated into a $\mathcal{SROIQ}$ knowledge base; for example, transitive properties cannot be used in number restrictions. Ontologies that satisfy these syntactic conditions are called OWL 2 DL ontologies. "OWL 2 DL" is used informally to refer to OWL 2 DL ontologies interpreted using the Direct Semantics.

The RDF-Based Semantics assigns meaning directly to RDF graphs and so indirectly to ontology structures via the mapping to RDF graphs. The RDF-Based Semantics is fully compatible with the RDF Semantics, and extends the semantic conditions defined for RDF. The RDF-Based Semantics can be applied to any OWL 2 Ontology, without restrictions, as any OWL 2 Ontology can be mapped to RDF. "OWL 2 Full" is used informally to refer to RDF graphs considered as OWL 2 ontologies and interpreted using the RDF-Based Semantics.

With the support of much tools, RDF and OWL 2 are becoming more and more popular in academic and industry application. In our research, OWL 2 is chosen to represent ontology in general. However, the standardization of OWL also leaves (at least) two crucial issues for Web-based ontologies unsatisfactorily resolved, namely how to represent and reason with multiple distinct, but linked ontologies, and how to enable effective knowledge reuse and sharing on the Semantic Web [50]. We will discuss more about this issue in the following sections.

2.4.5 Description Logics

In previous section, a brief but not complete introduction about ontology and relative technologies are presented. This section will introduce the underlying logics behind ontology language.

Description Logics came out of the efforts trying to improve knowledge representation in 1980's. At that time, approaches to knowledge representation are divided roughly into logic-based and non-logic based. Since first-order logic provides very powerful and general machinery, logic-based approaches were more general-purpose from the very start. In a logic-based approach, the representation language is usually a variant of first-order predicate calculus, and reasoning amounts to verifying logical consequence [51]. Description Logics (DL), or Description Logics family, according to its expressiveness, are decidable fragments of first order logic (FOL) [52]. A DL knowledge base is analogously typically comprised by two components — a "TBox" and an "ABox". The TBox contains intensional knowledge in the form of a terminology (hence the term "TBox," but "taxonomy" could be used as well) and is built through declarations that describe general properties of concepts. Because of the nature of the subsumption relationships among the concepts that constitute the terminology, TBoxes are usually thought of as having a lattice-like structure; this mathematical structure is entailed by the subsumption relationship — it has nothing to do with any implementation. The ABox contains extensional knowledge — also called assertional knowledge (hence the term "ABox") — knowledge that is specific to the individuals of the domain of discourse. Intensional knowledge is usually thought not to change — to be "timeless", in a way — and extensional knowledge is usually thought to be contingent, or dependent on a single set of circumstances, and therefore subject to occasional or even constant change [51].

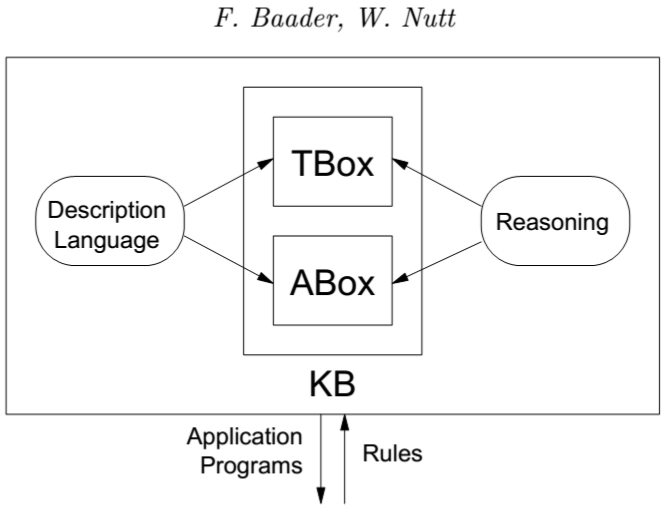


Figure 2.4: Architecture of a knowledge representation system based on Description Logics [1].

Most implemented DL systems provide for a rule language, which can be seen as a very simple, but effective, application programming mechanism. A DL system not only stores terminologies and assertions, but also offers services that reason about them. Typical reasoning tasks for a terminology are to determine whether a description is satisfiable (i.e., non-contradictory), or whether one description is more general than another one, that is, whether the first subsumes the second. Important problems for an ABox are to find out whether its set of assertions is consistent, that is, whether it has a model, and whether the assertions in the ABox entail that a particular individual is an instance of a given concept description[1].

Constructor	Syntax	Semantics
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value restriction	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y : (x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
at-least restriction	$(\geqslant nr.C)$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq n\}$
at-most restriction	$(\leqslant nr.C)$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}$

Figure 2.5: Syntax and semantics of concept descriptions in DL

Concept descriptions built from the constructors are shown in Figure 2.5, where C, D

stand for concept descriptions, r for a role name, and n for a nonnegative integer. In the description logic ALC (Attributive Concept Language with Complements), concept descriptions are formed using the constructors negation, conjunction, disjunction, value restriction, and existential restriction. The description logic $ALCQ$ additionally provides us with (qualified) at-least and at-most number restrictions [53]. Figure 2.6 is a TBox example of concepts and relations about family members. Usually, TBox is about abstract concepts, TBox has nothing to do with individuals or instances. On the other hand, facts and relations between individuals can be found in ABox.

$$
\begin{aligned}
\mathsf{Woman} &\equiv \mathsf{Person} \sqcap \mathsf{Female} \\
\mathsf{Man} &\equiv \mathsf{Person} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasChild.Person} \\
\mathsf{Father} &\equiv \mathsf{Man} \sqcap \exists\mathsf{hasChild.Person} \\
\mathsf{Parent} &\equiv \mathsf{Father} \sqcup \mathsf{Mother} \\
\mathsf{Grandmother} &\equiv \mathsf{Mother} \sqcap \exists\mathsf{hasChild.Parent} \\
\mathsf{MotherWithManyChildren} &\equiv \mathsf{Mother} \sqcap \geqslant 3\,\mathsf{hasChild} \\
\mathsf{MotherWithoutDaughter} &\equiv \mathsf{Mother} \sqcap \forall\mathsf{hasChild}.\neg\mathsf{Woman} \\
\mathsf{Wife} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasHusband.Man}
\end{aligned}
$$

Figure 2.6: A terminology (TBox) with concepts about family relationships [1].

2.4.6 Ontology Reasoning

For a very small ontology containing only several axioms and individuals, it is quite easy to assure the logic correctness by a reasonable mind. While, in practice, big ontologies equipped with thousands of concepts and individuals between which sophisticated relationships being waved are impossible to be verified correctly by human mind. To deal with this problem, logic reasoner was invented to make sure knowledge base is built in logic correct manner and internal concepts as well as individuals are consistent with each other.

As for the work of a reasoner, checking satisfiability of concepts is a key inference. As we can see, a number of other important inferences for concepts can be reduced to the (un)satisfiability. For instance, in order to check whether a domain model is correct, or to optimize queries that are formulated as concepts, we may want to know whether some concept is more general than another one: this is the subsumption problem [1]. A concept C is subsumed by a concept D if in every model of T the set denoted by C is a subset of the set denoted by D. Algorithms that check subsumption are also employed to organize the concepts of a TBox in a taxonomy according to their generality. Further interesting relationships between concepts are equivalence and disjointness.

These properties are formally defined as follows. Let T be a TBox.

Satisfiability: A concept C is *satisfiable* with respect to $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}$ is nonempty. In this case we say also that $\mathcal{I}$ is a *model* of C.

Subsumption: A concept C is *subsumed* by a concept D with respect to $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$. In this case we write $C \sqsubseteq_{\mathcal{T}} D$ or $\mathcal{T} \models C \sqsubseteq D$.

Equivalence: Two concepts C and D are *equivalent* with respect to $\mathcal{T}$ if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$. In this case we write $C \equiv_{\mathcal{T}} D$ or $\mathcal{T} \models C \equiv D$.

Disjointness: Two concepts C and D are *disjoint* with respect to $\mathcal{T}$ if $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for every model $\mathcal{I}$ of $\mathcal{T}$.

Figure 2.7: Four main inference tasks of DL reasoning [1].

Traditionally, the basic reasoning mechanism provided by DL systems checked the subsumption of concepts. This, in fact, is sufficient to implement also the other inferences, as can be seen by the following reductions. For concepts C, D we have (i) C is unsatisfiable $\Leftrightarrow$ C is subsumed by $\bot$; (ii) C and D are equivalent $\Leftrightarrow$ C is subsumed by D and D is subsumed by C; (iii) C and D are disjoint $\Leftrightarrow$ $C \sqcap$ is subsumed by $\bot$ [1].

All description languages implemented in actual DL systems provide the intersection operator "$\sqcap$" and almost all of them contain an unsatisfiable concept. Thus, most DL systems that can check subsumption can perform all four inferences defined above.

If, in addition to intersection, a system allows one also to form the negation of a description, one can reduce subsumption, equivalence, and disjointness of concepts to the satisfiability problem. As in the case of consistency, reasoning tasks for ABoxes with respect to acyclic TBoxes can be reduced to reasoning on expanded ABoxes [1]. Tableau algorithms are effective reasoning algorithms that are being used in various DL reasoners. Tableau algorithms focus on satisfiability problems; other reasoning problems, like sub-sumption, or entailment (the main inference problem in OWL) can be solved by reducing them to satisfiability problems. Tableau algorithms for expressive DLs are non-deterministic in the sense that there might exist completion rules that yield more than one possible outcome. A tableau algorithm will return "satisfiable" iff there exists at least one way to apply the non-deterministic rules such that a clash-free graph is obtained, to which no rules are applicable. Termination is guaranteed through blocking: halting the expansion process when a "cycle" is detected [53]. When the algorithm detects that a path in the graph will be expanded forever without encountering a contradiction, then the application of the generating rules is blocked, so that no new nodes will be added to that path. Investigating this trade-off between the expressivity of DLs and the complexity of their inference problems has been one of the most important issues in DL research [53].

2.4.7 Ontology Engineering

The discipline that investigates the principles, methods and tools for initiating, developing and maintaining ontologies is Ontology Engineering [54]. As ontology is getting bigger and more complex, we certainly need formal and systematic engineering methodology to guide and manage the ontology development. As ontologies are quite subjective artifacts, knowledge, experience or even personal emotion and preference may influence what and how ontology is built. To ensure the quality of ontology development, quite several ontology engineering methodologies are proposed. Some good practice and wisdom from software engineering are reflected in these methodologies. However, according to [42], there is no one "correct" way or methodology for developing ontologies. There are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.

In the same guide [42], a process of interactive design & A Simple Knowledge-Engineering Methodology is presented:

- Step 1 Determine the domain and scope of the ontology
- Step 2 Consider reusing existing ontologies
- Step 3 Enumerate important terms in the ontology
- Step 4 Define the classes and the class hierarchy (top-down, bottom-up, combination)
- Step 5 Define the properties of classes—slots
- Step 6 Define the facets of the slots
- Step 7 Create instances

Early in [55], an evolving prototype as the ontology life cycle is proposed. The life of an ontology moves through: specification, conceptualization, formalization, integration, implementation, and maintenance. The evolving prototype life cycle allows the ontologist to go back from any state to other if some definition is missed or wrong. So, this life cycle permits the inclusion, removal or modification of definitions anytime of the ontology life cycle. The reuse of existing ontologies is highly recommended.

In [56], a more complete and systematic ontology engineering methodology called DILIGENT is proposed. There are different kinds of participants in the DILIGENT process: (1) domain experts, that know about the domain that is targeted (2) ontology engineers, that know how to build ontologies (3) knowledge engineers, that know how to build knowledge or information systems based on ontologies, and (4) users, that use the ontology resulting from the process in their systems for their own uses. The participants directly involved in building the ontology, may or may not use the ontology. However, most ontology users will typically not build or modify the given ontology.

The process comprises five main activities (Figure 2.8): (1) build, (2) local adaptation, (3) analysis, (4) revision, (5) local update. The process starts by having domain experts, users, knowledge engineers and ontology engineers building an initial ontology. The team involved in building the initial ontology should be relatively small, in order to more easily find a small and consensual first version of the shared ontology. At this point, it is not required to arrive at an initial ontology that covers the complete domain.

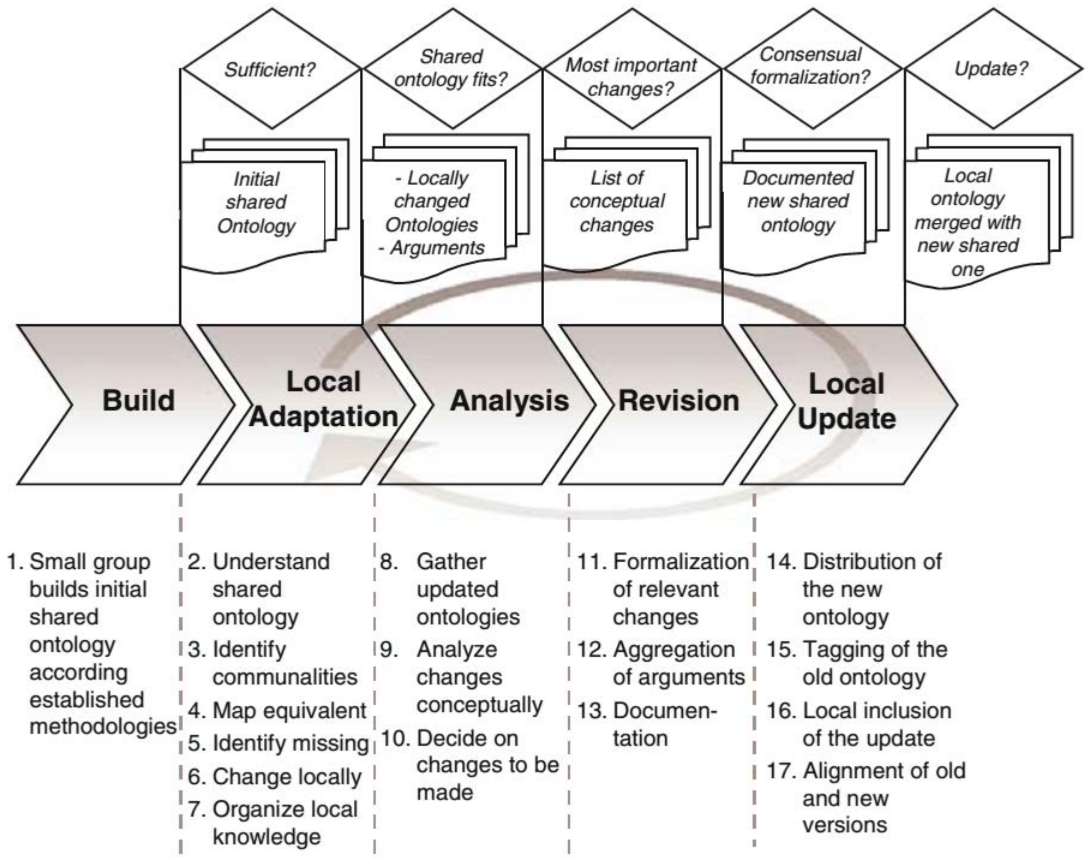


Figure 2.8: DILIGENT process stages (1-5), actions (1-17) and structures

In DILIGENT there are two kinds of ontologies: The shared ontology and local ontologies. The shared ontology is available to all users and cannot be changed directly except by the board. Users are free to change, in their local environments, a copy of the shared ontology. The ontology resulting from the changes of a user is the user local ontology.

In contrast to other approaches that provide methodological guidance for ontology engineering, the NeOn [57] Methodology does not prescribe a rigid work-flow, but instead it suggests a variety of pathways for developing ontologies. NeOn methodology includes the following components:

- The NeOn Glossary, which identifies and defines the processes and activities potentially involved in the ontology network construction. There are 10 processes are defined, e.g.

ontology aligning, ontology design pattern reuse, non-ontological resource reuse, etc; 48 activities are identified, e.g. ontology annotation, ontology assessment, ontology comparison, ontology modularization, etc.

- A set of nine scenarios for building ontologies and ontology networks (From specification to implementation; Reusing and re-engineering non-ontological resources; Reusing ontological resources; Reusing and re-engineering ontological resources; Reusing and merging ontological resources; Reusing, merging, and re-engineering ontological resources; Reusing ontology design patterns; Restructuring ontological resources; Localizing ontological resources.) Each scenario is decomposed in different processes and activities taken from those included in the NeOn Glossary.

- Two ontology network life cycle models (Water-fall model and Iterative-Incremental model) that specify how to organize the processes and activities of the NeOn Glossary into phases.

- A set of prescriptive methodological guidelines for processes and activities.

[58] describes a gap between the ontology engineering tools and the traditional software engineering in Semantic Web development. The primary objective of their work is to bridge this gap between two different, but complementary engineering disciplines with a systematic approach. This approach leverages Model-Driven Architecture (MDA) and Ontology Definition Metamodel (ODM), which enable model transformation. This approach allows seamlessly supporting existing models in UML and other languages in Semantic Web-based software development. In addition, it allows exploiting the availability and features of UML tools for creation of vocabularies and ontologies. The basic idea of MDA is that the system functionality is defined as a platform independent model, using an appropriate specification language and then translate to one or more platform-specific models for the actual implementation. According to our understanding, ontology won't be built directly, instead, existent UML model or other model are used as input to transform them into ontology. This method is quite appealing in terms of interoperability.

A Human-Centered Ontology Engineering Methodology (HCOME) for the development and evaluation of living ontologies in the context of communities of knowledge workers. HCOME provides clear distinction between the phases of the ontology life cycle, goals that should be achieved in each phase and task that can be performed so as to achieve these goals. These are summarized in Figure 2.9. As it is shown, these tasks are performed iteratively, until a consensus has been reached between knowledge workers. Tasks are performed by workers either individually or conversationally. In the first case, we consider that tasks are performed in the personal space of workers (marked with the letter "P"). In the latter case, tasks are performed in an information space that a group of knowledge workers share, i.e. in a shared space (marked in Table 1 with the letter "S"). A worker can initiate any ontology engineering task in his personal or shared space, or take part to a task that has been initiated by other members of the community.

Ontology life cycle phases	Goals	Tasks
Specification	Define aim/scope/ requirements/teams	Discuss requirements (S)
		Produce documents (S) Identify collaborators (S)
Conceptualisation	Acquire knowledge	Import from ontology libraries **(P)** Consult generic top ontology **(P)** Consult domain experts by discussion (S)
	Develop and maintain ontology	Improvise **(P)**
		Manage conceptualisations **(P)** Merge versions **(P)** Compare own versions **(P)** Generalize/specialize versions **(P)** Add documentation **(P)**
Exploitation	Use ontology	Browse ontology **(P)** Exploit in applications
	Evaluate ontology	Initiate arguments and criticism (S) Compare others' versions (S) Browse/exploit agreed ontologies (S) Manage the recorded discussions upon an ontology (S) Propose new ontology versions by incorporating suggested changes (S)

Figure 2.9: The HCOME methodology phases to ontology engineering.

In summary, ontology engineering has aroused the interest of researchers and drawn the attention of the industry. However, compared to software engineering, ontology engineering is still in its infancy, in particularly, there is still a lack of advanced tool support for various ontology engineering activities.

2.4.8 Modular Ontology

What is modular ontology or ontology modularity? What is an ontology module? According to [59]: *an ontology module is a reusable component of a larger or more complex ontology, which is self-contained but bears a definite association to other ontology modules, including the original ontology.* In [60], the author gives another definition: *A module is a subset of an ontology that captures all the knowledge the ontology contains about a given set of terms.*

More specifically, we consider an ontology O as a set of axioms (subclass, equivalence, instantiation, etc.) and the signature Sig(O) of an ontology O as the set of entity names occurring in the axioms of O, i.e. its vocabulary [61]. Then a ontology module has definition like this:

Let L be a description logic, $Q1 \subseteq Q$ be two ontologies expressed in L and let S be a signature. We say that $Q1$ is an S-module in Q w.r.t. L, if for every ontology P and every *axiom* α *expressed in* L *with* $Sig(P \cup \alpha) \cap Sig(Q) \subseteq S$, *we have* $P \cup Q \models \alpha$ *iff* $P \cup Q1 \models \alpha$. [62]

There is currently an important growth in interest concerning modularization techniques for ontologies, as more ontology designers and users become aware of the difficulty of reusing, exploiting and maintaining big, monolithic ontologies. The considered notion of modularity comes from software engineering, but, unfortunately, it is not yet as well understood and used in the context of ontology design as it is for software development [63]. In general, there are two contexts identified with the idea of ontology modularization. One is ontology integration or interrelation, which means multiple ontology modules are put together to compose a new ontology. The other one is about module extraction and module partition. Our research in Chapter 3 will introduce some experience which has something to do with module partition. Modularity is also an interesting approach that deals with ontology reusability, which reminds us of similar scenarios in software engineering. Modularization materializes the long-established complexity management technique known as divide and conquer. It is routinely used in various areas of computer science, such as algorithms, software engineering, and programming languages. In software engineering, for example, module is one of the terms used to denote a software component that is designed to perform a given task and is intended to interact with other modules within a larger software architecture. In programming languages, module sometimes denotes an encapsulation of some data. Easiness of understanding and potential for reusing are among the main claimed benefits of these approaches. According to [63] again, "The compositional approach to modular ontologies relies on appropriate definitions of interfaces between different modules to be connected. In an ideal case, these interfaces are defined at design time when modules are created in a modular fashion. In reality, we are faced with a situation where no interfaces are defined and relevant connections between ontologies have to be discovered and represented at composition time. There are two main lines of research addressing this problem. The first line is concerned with the development of methods for identifying semantic relations between elements in different ontologies. The second line of research is concerned with formalisms for encoding and using semantic relations (mappings) between ontologies. These formalisms are often based on non-standard extensions of the logics used to encode the ontologies." Our work introduced in Chapter 4 has actually followed the second line.

[63] has also specified the goals of ontology modularization: Scalability for querying data and reasoning on ontologies; Scalability for evolution and maintenance; Complexity management; Understandability; Context-awareness and Personalization; Reuse. An arithmetic metaphor is also given like this: module = a (smaller) ontology + inter-modules links. In our research, we will take these goals into consideration and try to realize them.

Several modularization technologies were developed to combine ontologies, while there are still few methods to divide modularization in or retrieve modules from bigger ontology. A survey of modular ontology technology was investigated in [64], criteria such as networking, dynamics, distribution and reasoning were used to inspect each formalism or technology. The comparison result suggests that no existing approach could satisfactorily meet the requirement of networked ontology applications.

An original mechanism of OWL dedicated to address ontology reuse and modularity is owl:imports syntax. Critics about this syntax can be found in [65] where the authors stated that OWL axioms and entities are always global and can affect any entity in either importing or imported modules. In [2], the authors criticize that the semantics of OWL is defined in such a way that all ontologies, imported and importing, share the single global interpretation. As a consequence, imported and importing ontologies cannot describe differently the very same portion of the world, using different perspectives and levels of granularity, without rising a logical contradiction. *Owl:imports* concerns with its ability to import ontologies only as a whole, while it is often the case in practice that only a certain part of imported ontology is of interest to the importing ontology. Partial import is not supported.

E-Connections were originally introduced as a way to increase the expressivity of each of the component logics, while preserving the decidability of the reasoning services. Thus, E-Connections were conceived for providing a trade-off between the expressivity gained and the computational robustness of the combination. An E-Connection is a set of “connected” ontologies. An E-Connected ontology not only contains information about classes, properties and their instances, but also about a new kind of properties, called link properties, which establish connection between the ontologies.

An E-Connected ontology can be roughly described as an OWL-DL ontology, extended with the ability to define link properties and construct new classes in terms of restrictions on them. From the modeling perspective, each of the component ontologies in an E-Connection is modeling a different application domain, while the E-Connection itself models the union of all these domains. For example, an E-Connection could be used to model all the relevant information referred to a certain university, and each of its component ontologies could model, respectively, the domain of people involved in the university, the domain of schools and departments, the domain of courses, etc [66].

Another interesting method is DDL (Distributed Description Logics) framework, C-OWL and the distributed reasoner DRAGO(Distributed Reasoning Architecture for a Galaxy of Ontologies). This method considers construction of modular ontologies from purely terminological $SHIQ$ ontologies. Such a restriction is explained by the current limitations of the DDL distributed reasoning technique implemented in DRAGO. Practically, DRAGO works with ontologies represented in OWL and semantic mappings encoded in C-OWL. New syntax called *Bridge rules* are introduced to represent various relationships between entities that come from different ontologies. A contextual ontology is therefore the pair: OWL ontology, set of C-OWL mappings, where each C-OWL mapping is a set of bridge rules with the same target ontology. The C-OWL mappings are materialized in a XML representation. Due to the restrictions of the distributed reasoning algorithm to $SHIQ$ ontologies, DRAGO prohibits the use of nominal-related constructs in OWL, such as *owl:oneOf* and *owl:hasValue* [2]. According to DDL, a modular ontology is formally encoded into a distributed T-box, comprising a set of T-boxes (one for each ontological module), which are pairwise interrelated by “bridge rules” (inter-module connectives allowing to access and import knowledge contained in modules). The semantics of DDL fits into requirements of semantic locality and directionality, the partial reuse criterion is met due to the use of bridge rules allowing to selectively access the knowledge in the modules, loose coupling and reasoning scalability is met by the use of the distributed reasoning approach of DDL.

In Figure 2.10, T_i and T_j are distributed T-boxes, B_{ij} is a set of bridge rules mapping concepts from T_i to T_j. In accordance with the definition of interpretation in Description Logic, each interpretation $I_i(I_j)$ consists of a non-empty, possibly infinite domain Δ^{I_i}, and a valuation function $\cdot^{I_i}$. A domain relation r_{ij} from Δ^{I_i} to Δ^{I_j} is a subset of $\Delta^{I_i} \times \Delta^{I_j}$. The syntax and semantics of bridge rules are vital to DDL's modeling and reasoning.

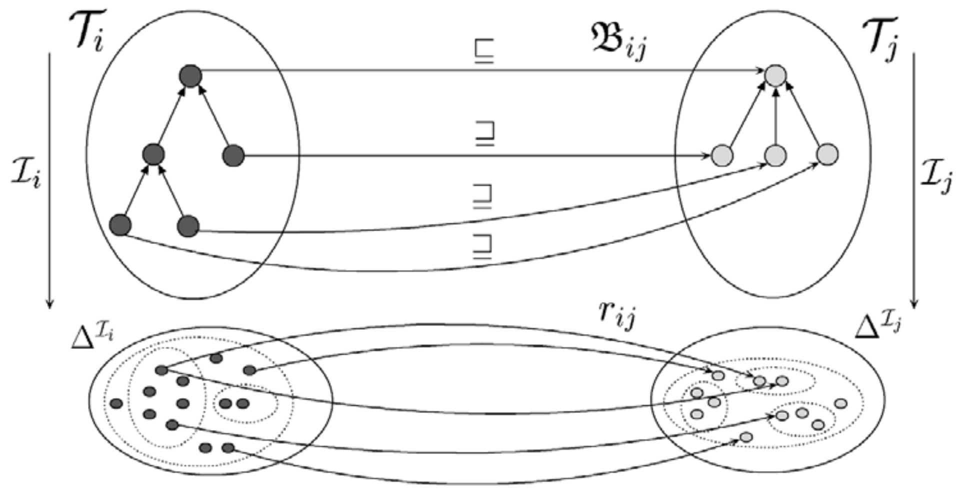


Figure 2.10: Visualized semantics of DDL [2].

In DDL, a combination of onto- and into-bridge rules allows for propagating subsumption axioms across ontologies. Practically, this means that if an ontology T_1 contains an axiom **InBook** $\sqsubseteq$ **Publication**, and an ontology T_2 just defines two concepts **ScientificPaper** and **BookArticle**, then bridge rules $1:$ **Publication** $\xrightarrow{\sqsubseteq}$ $2:$ **ScientificPaper** and $1:$ **In-Book** $\xrightarrow{\sqsupseteq}$ $2:$ **BookArticle** entail in T_2 the axiom that **BookArticle** is a **ScientificPaper**. Figure 2.11 illustrates this simple subsumption propagation graphically. In languages that support disjunction, the simplest propagation rule can be generalized to the propagation of subsumption between a concept and a disjunction of other concepts [3].

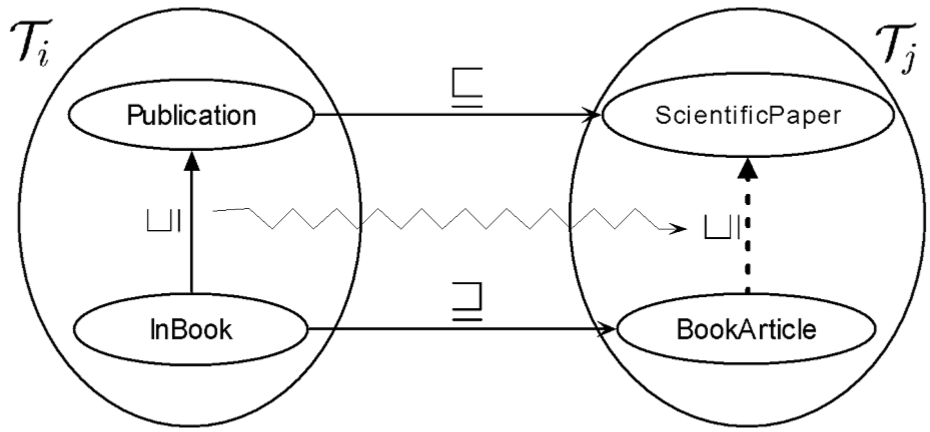


Figure 2.11: Simple propagation of a subsumption axiom from T_i to T_j [3].

DDL and E-Connections are motivated by, and hence are responsive to, different application scenarios. Their expressivity and reasoning power is complementary in several ways. However, both of them are also limited in several ways. Due to the strong local domain disjointness assumption adopted in those approaches, the distributed reasoning processes with such approaches may encounter semantic difficulties. They also fail to provide solutions for some critical distributed reasoning tasks, such as transitive concept subsumption across multiple modules. It is stated in [67] that, to preserve contextuality, existing modular ontology languages offer only limited ways to connect ontology modules and, hence, limited ability to reuse knowledge across modules. For instance, DDL does not allow concept construction using foreign roles or concepts. E-Connections, on the other hand, does not allow concept subsumptions across ontology modules or the use of foreign roles. Finally, Semantic Importing, in its current form, only allows each component module to be in ALC. None of the existing approaches supports knowledge reuse in a setting where each ontology module uses a representation language that is as expressive as OWL-DL, i.e., $SHOIN(D)$.

P-DL (Package-Based Description Logics) [68], by relaxing the local domain disjointness assumption, allows discovery of a distributed model for a set of ontology modules that is identical to that obtainable by combining the ontology modules into a centralized ontology, a property we refer to as exactness of distributed reasoning relative to its centralized counterpart. P-DL uses importing relations to connect local models. In contrast to OWL, which forces the model of an imported ontology be completely embedded in a global model, the P-DL importing relation is partial in that only commonly shared terms are interpreted in the overlapping part of local models. Thus, a P-DL model is a virtual model constructed from partially overlapping local models by merging "shared" individuals, as shown in Figure 2.12 [4].

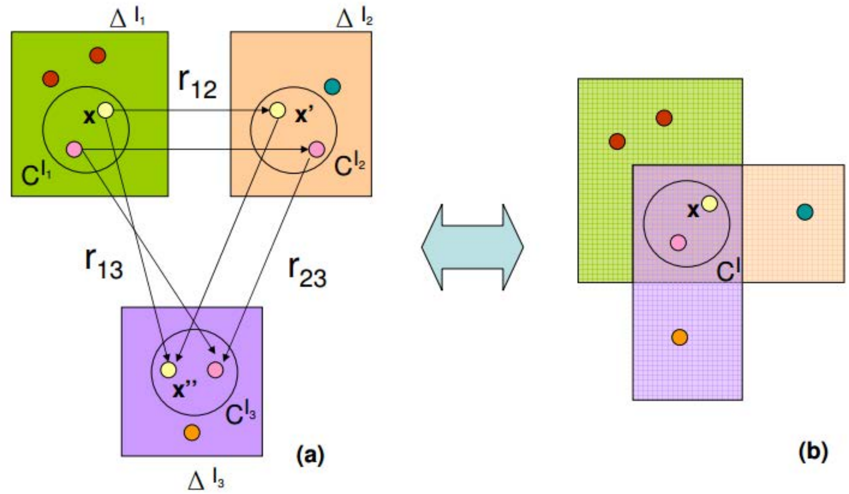


Figure 2.12: P-DL Semantics (a) partially overlapping local models (b) virtual global model [4].

[68] also investigates a sound and complete tableau-based reasoning algorithm for a P-DL language $ALCPCc$, which extends ALC with acyclic concept importing between packages. The algorithm allows the reasoning process to be distributed based on local reasoning services offered by each module. Local tableaux associated with the ontology modules while physically separate, may conceptually overlap by communicating with each other via a set of messages. Our preliminary investigation shows that the proposed algorithm can solve many known reasoning difficulties in existing approaches. Complexity study shows the algorithm for the package-based ALC language has the same time complexity as that of a canonical ALC reasoning algorithm [69].

While these methods and formalisms more or less set up a logics syntax barricade that could limit a large scale reasoning and modification between heterogeneous and distributed modular ontologies, e.g. the underlying logics formalism of E-Connection is OWL-DL (i.e. $SHOIN$); while, logics formalism for DDL is $SHIQ$; when it comes to Package-Based Description, it turns into $SHOIQ$. According to different survey research, there are quite a number of other modular ontology approaches or distributed ontology languages available. In [70] and [64], survey and comparison between different modular ontology formalisms are presented. Synthesizing the result from the both, we have a status overview for several popular formalisms as illustrated in Table 2.2. Encapsulation means modules are autonomously created and maintained in the domain of question. The ontology dynamics of networked ontologies reflects the importance of monitoring and propagating the ontology changing events, especially when some ontology modules are changed and updated. Loose Coupling indicates that existence of overlaps between the modules is uncertain, the ontologies should be compatible while being composed. In other words, formalism for modular ontologies should be able to handle modules that are not disjoint with each other. Self-Containment means formalism for modular ontologies should support localized semantics that include the global information about module dependencies.

Table 2.2: An comparison of several popular modular ontology formalisms (T stands for TBox and A for ABox)

	OWL DL	DDL	IDDL	P-DL	E-Connection
Encapsulation	Partial	Yes	Yes	Yes	Yes
Re-usability	Fair	Good	Good	Good+	Good-
Trust and Security	No	No	No	Partial	No
Ontology Dynamics	Yes	No	No	No	No
Loose Coupling	No	Yes	Yes	Yes	Yes-
Self-Containment	Partial	Yes	Yes	Partial	Yes
Scalability	Low	Fair	Fair+	Fair	Low
Reasoning Support	T and A	T and Partial A	T and A (under investigation)	T	T
Context Awareness	No	Yes	Yes	Yes	Yes
Heterogeneity Robustness	Very Limited	Good	Very Good	Limited	Excellent

From a practical perspective, these methods have not been applied in such a considerable scope that we can have successful application cases for a good study. As stated in [4], a consensus on an OWL-compatible syntax for a modular ontology language that can express both inter-module concept subsumptions and inter-module role relations is still lacking. It would be interesting to investigate whether OWL can be re-modeled with a new modular semantics, or it has to be extended with a new set of constructors to replace $owl : imports$. Mature tools for building modular ontologies are needed. [71] has also showed the need to enrich ontology with the capability to cope with: directionality of information, we need to keep track of the source and the target ontology of a specific piece of information; local domains, we need to give up the hypothesis that all ontologies are interpreted in a single global domain; context mappings, we need to be able to state that two elements (concepts, roles, individuals) of two ontologies, though being (extensionally) different, are contextually related, for instance because they both refer to the same object in the world. In [72], authors introduced some other topics e.g. Loose Coupling: In general, we cannot assume that two ontology modules have anything in common. This refers to the conceptualization as well as the specific logical language used for the interpretation of objects, concepts, or relations; Self-Containment: In order to facilitate the reuse of individual modules we have to make sure that modules are self-contained. In particular, the result of certain reasoning tasks such as subsumption or query answering within a single module should be possible without having to access other modules. We try to include such issues that have been discussed above in our research.

2.5 Conclusion

In this chapter, we have mainly introduced the problematics and context of our research from a scientific point of view. We start from decision support system for eco-labeling and find that there is no much decision support application or practice aiming at eco-labeling. Most of the certification and evaluation work concerning eco-labeling is accomplished manually.

However, in a more general context, decision support based on ontology knowledge base has drawn some attention and some cases can be traced. So, our work fills this gap between decision support and eco-labeling. In our research, we will continue to explore how to make use of ontology related technologies to explicitly build better decision support systems, furthermore, exploit semantic inference to generate explanation for decision making. In the third section of this chapter, a current status of ontology, ontology engineering, and modular ontology is introduced. We have learned that ontology is a critical component of Semantic Web which is an appealing vision of more powerful Internet in the future. RDF and OWL are key technologies that implement linked data and ontology. Description Logics is the logics basis for OWL. As ontology is becoming larger and distributed, modular ontology is becoming more and more important as for ontology engineering. In order to implement decision support system based on ontology knowledge base, the first step is the development of ontology. In the next chapter, we will introduce our approach to develop modularized ontology, and how to apply the entity-rule separation pattern to realize better modularity and re-usability.

Chapter 3

A knowledge base based on modular ontology

3.1 Introduction

Nowadays, most of the knowledge and criteria about eco-labeled products is published in official journals, web pages, and all kinds of documentation. Usually, these knowledge is presented in such complex regulation and specification document that it is difficult to be understood even by humans. Integrating these criteria in software systems in order to popularize eco-labels and accelerate eco-labeling process requires that these knowledge must be exploitable by machines. However, until now, there is still a lack of computable format of them. Besides, traditional knowledge base in relational data model is not inter-operable, lacks of inference support and is difficult to be reused. In order to better understand these rules and criteria, stakeholders need a common and machine accessible presentation of knowledge. To address such problems, in our research, we propose an ontological knowledge base composed of modularized ontologies. This knowledge base scheme has been illustrated and applied to the creation of EU Eco-label's laundry detergent products' criterion ontology, and laundry detergent product has been chosen as our study case.

3.2 State of art

Very lately, with the rise of "Cloud", "Big Data" and "Linked Data"[1], the data support for decision making is becoming enormous in quantity and more intelligent function for data analysis is required. More powerful distributed calculation and more frequent data transmission require cost control as for the communication between software agents. Also, today's enterprise and organization seek for cooperation on higher level, requiring that information

[1] http://www.w3.org/standards/semanticweb/data

exchange and communication should not only stay on data level, but also on application and business levels, in other word, a semantic interoperability of information sharing.

Traditional application architecture treats data like some goods circulating in the assembly line or some consumable resources. The processing logic is weaved in the computer program. To solve interoperability problem, ontology is suggested to extract knowledge out of program logics and make specific knowledge stand alone. In such way, knowledge and logic with semantics could be exchanged and shared.

Derived from philosophy though, in computer science, we refer to an ontology as a special kind of information object or computational artifact [73]. Studer et al. [74] gave definition stating that: "An ontology is a formal, explicit specification of a shared conceptualization." Today, so many ontologies and knowledge repositories have been developed and adapted into applications, especially in biomedical domains [75]. Successful examples and platforms are BioPortal [2], UniProt[3], LEO [4], etc.

Traditional database based on relationship model is becoming clumsy, especially in data exchange and inference aspects. Ontology technologies have attracted much attention. People are seeking for new and hybrid technologies to construct more powerful software tools. [76] describes a project called SEEK (Science Environment for Ecological Knowledge) that developed a collection of ontologies for describing ecological organisms, systems, and observations. The two major uses of ontologies are accessing and analyzing ecologically important information. In industry domain, some pioneer and successful applications also came up. For example in automotive industry, a new site search and browse engine based on ontology and semantic annotation is proposed in [77]. Traditional web resources are semantically annotated with RDFa using vocabularies in Car Options Ontology[5], Volkswagen Vehicles Ontology[6] and Good Relations[7]. In [32], OWL/SWRL[8] modelling paradigm representing EDF MUDU(French acronym for user data unified model) is developed. This ontology model is made of a set of coherent user dictionaries that are used to generate business catalogues for each nuclear power plant project and each business activity. In [78] a product modeling ontology called PRONTO is proposed and practiced in food industry. This proposal tries to meet the need for an integrated product model to be shared by all the organizations participating in global supply chains or all areas within a company. It has also been proved that document search as well as passage search, knowledge base search using SPARQL[9] on RDF[10] triple stores are effective for Question-Answering(QA) system [79].

Despite quite amount of ontologies of different domains have been developed, lots of

[2]http://bioportal.bioontology.org/
[3]http://www.uniprot.org/
[4]http://leo.informea.org/
[5]http://www.volkswagen.co.uk/vocabularies/coo/ns
[6]http://www.volkswagen.co.uk/vocabularies/vvo/ns
[7]http://www.heppnetz.de/projects/goodrelations/
[8]https://www.w3.org/Submission/SWRL/
[9]https://www.w3.org/TR/rdf-sparql-query/
[10]https://www.w3.org/RDF/

problems are encountered when knowledge engineers as well as general users want to understand and reuse the ontologies into their own development. One reason of such difficulties is the semantic confusion and heterogeneity between domains. As for application of ontology, there is definite need to gather knowledge from multiple remote ontological sources. It is known that, when knowledge is distributed, the idea to collect all knowledge into a single repository (i.e. the integration approach) is very difficult to be implemented, because semantic heterogeneity calls for human processing [80]. Another very important reason is the low reusable design of these ontologies. Good ontology design pattern has drawn the attention of many researchers. In [81] and [82], a method to describe ontology design pattern is presented. A Semantic Web portal called OntologyDesignPatterns.org[11] is also available. However, most of the submitted patterns are cataloged in Content ODPs which means that the patterns themselves may contain certain semantics and domain knowledge, which may still set obstacles to reuse. Thus, higher level engineering principle and philosophy is needed.

An interesting approach that deals with ontology reusability is modularity. Generally speaking, there are two important aspects of ontology modularization: independently developing modules that can be integrated coherently and uniformly (ontology composition) or extracting such modules from an integrated ontology for supporting a particular use case (ontology decomposition) [75]. Most of our research focus on the first aspect and we emphasize more on reuse, inference and change management of ontology knowledge base.

To achieve ontology modularity in a distributed scenario, different methods and schemes are proposed. For example, E-Connection is proposed as a set of "connected" ontologies. An E-Connected ontology contains not only information about classes, properties and their individuals, but also a new kind of properties, called Link Properties, which establish the connection between the ontologies [83]. Another interesting approach is Distributed Description Logics (DDL) framework [2] and the distributed reasoner DRAGO (Distributed Reasoning Architecture for a Galaxy of Ontologies) [84] as formal and practical tools for composing modular ontologies. Also, we have Package-Based Description Logics as another formalism that supports contextual reuse of knowledge from multiple ontology modules [68]. While, these methods and formalisms have more or less logics compatibility problems when we try to use them together.

From practical perspective, these methods have not been applied in such a considerable scale and this lack of practical and mature engineering experience is also a problem. As for OWL ontology, the current OWL *imports* syntax already provides the ability of modularization to a certain extent. In this chapter, we apply a method using *imports* syntax to build OWL ontology knowledge base with SWRL rules in which smaller ontology components can be maintained and reused more easily. We expect to explore and find out some useful design principles or engineering experience with regards to the original OWL ontology scheme.

Like software engineering, engineering methodologies are also required in ontology development. Yet, in the authors' eyes, ontology engineering is not as mature as software engineering because of its shorter history and relative limited scale of practice. In spite of that,

[11] http://ontologydesignpatterns.org/wiki/Main_Page

quite several ontology development methods have been proposed, e.g. TOVE, METHONTOLOGY, DILIGENT, NeOn Methodology [85][86][54][57]. Most of these methods follow a "water fall" pattern. Common characteristics that can be generalized from these methods are iteration and refinement. In our ontology development, we don't rely on unique methodology exclusively, instead, we have adapted and customized those useful steps from all these methodologies to have a development method that best suit the task. The key steps in our development method are: requirement analysis, capture of motivating scenarios and competency questions, terminology collection, modeling, test reasoning and argumentation, evaluation and analysis. The rest part of this chapter will describe these steps and present the modularized ontologies in details.

3.3 Requirement analysis, motivating scenarios and competency questions

Firstly, let's have a brief overview of the current eco-labeling process for laundry detergent products. As EU Eco-label has been undergoing for more than twenty years in European Union, a well-defined coordination between the EU Commission and other member countries' competent bodies has been established. On the official web site of EU Eco-label[12], detailed documentation is provided to enterprises to facilitate the application process. On the same site, there is also a detailed product group catalog and corresponding criteria for each product or service group.

Usually, when a new product or service is about to be added into the product group catalog, stakeholders and domain experts will be assembled. After careful survey and discussion, a technical report will be drafted. According to this technical report, a feasible criteria will be made and then put into practice under the authorization of EU commission. From time to time, necessary revise or amendment to the criteria may be applied. As a result, the information implied in each product or service criteria becomes a complex knowledge system which involves multiple domains' expertise, standards and best practice. Take laundry detergent for example, criteria is set for each of the following aspects:

1. Dosage requirements.
2. Toxicity to aquatic organisms: Critical Dilution Volume (CDV).
3. Biodegradability of organics.
4. Excluded or limited substances and mixtures.
5. Packaging requirements.
6. Washing performance (fitness for use).

[12]http://ec.europa.eu/environment/ecolabel/how-to-apply-for-eu-ecolabel.html

7. Points.

8. Consumer information.

9. Information appearing on the EU Eco-label.

This criteria has been published in *Commission decision of 28 April 2011 on establishing the ecological criteria for the award of the EU Eco-label for laundry detergent 2011/264/EU*.[13] This commission decision is composed of regulation articles, annex where each item of the criteria is explained, and appendix. The regulation articles are not very interesting as it gives only administrative declaration and reference. Most of the knowledge about laundry detergent is elaborated in the annex and appendix. Criterion *"Dosage requirements"* specifies the reference product dosage recommended for each wash. Qualified detergent product should not exceed certain value. *"Toxicity to aquatic organisms"* specifies the maximum CDV value for qualified product. Similarly, in the next criterion *"Biodegradability of organics"*, it indicates that the content of organic substances in the product that are aerobically non-biodegradable (not readily biodegradable) (aNBO) and/or anaerobically non-biodegradable (anNBO) shall not exceed certain limits. Criterion *"Excluded or limited substances and mixtures"* prohibits some sensitive or hazardous substances as ingredients. *"Packaging requirements"* point out acceptable threshold weight/utility ratio (WUR) of the product. *"Washing performance"* is more about the product's performance test. The applicant shall provide a test report indicating that the product fulfills the minimum requirements specified in the test. Criterion *"Points"* provides an indicator matrix of points. Each option has 1 or 2 points. A minimum of 3 points shall be achieved for qualified product. Criterion *"Consumer information"* is totally subjective as it examines if the dosage instruction, washing recommendation, or pre-treatment information are properly printed on product's package. The last criterion *"Information appearing on the EU Eco-label"* is about the optional text showing on the EU Eco-label.

After reading and analyzing the criteria document for laundry detergent product, we have identified two important motivating scenarios or basic requirements concerning our ontology knowledge base. The first one is saving candidate product's detailed description. For example, some applicant wants his product to be eco-labeled, a description of the product should be provided. Product's critical physical and chemical characteristics, parameters, textual information or other specification should be instantiated in the ontology and can be queried afterwards. In other words, the ontology should not only be a knowledge base. More technically speaking, both TBox and ABox should be preserved in the ontologies. The other important motivating scenario is judging whether some candidate product is qualified to be labeled or not. This scenario requires inference support for the ontology.

Based on these two scenarios, some competency questions have been defined. We expect the ontology to be developed can answer questions like:

CQ1: If this product is qualified to be eco-labeled?

[13]http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32011D0264

CQ2: What is the quantitative value of this product's certain physical or chemical characteristics? (Critical dilution volume, biodegradability, weight/utility ratio, etc.)

CQ3: Does this product contain excluded or limited substances and mixtures?

CQ4: In which countries is this product being sold?

CQ5: What is the reference dosage per wash for this product?

CQ6: What is the corresponding EU Risk Phrase for some GHS Hazard Statement?

CQ7: What physical or chemical characteristics does some ingredient have? What are their values?

......

One thing that draws our attention is that, among those 9 criterion, some are not suitable to be modeled in ontologies. For example, the specification of consumer information has almost no quantitative parameter's requirement, instead, whether the information showing on the package is good or not is mostly subject to the judgment of human experts. As for the washing performance, a test report is needed, which must be carried out by certificated laboratory and then reviewed by human experts too. In our research, we had expected our knowledge base to cover all the criterion, but we found that some complex criterion are difficult to be translated and implemented in ontology. Because both the syntax and semantic complexity of this criterion has exceed what OWL ontology has. For example the criterion of points, it is required to calculate the points that a candidate product accumulates. With regard to OWL 2 and SWRL which are monotonic in terms of logics, it is hard to modify an already built model or to do calculation by itself. If we translate such kind of criterion into OWL ontology forcefully, we may encounter very strange and bulky ontology structure and that may cause extra complexity for the knowledge engineer and the reasoner. Thus, for the sake of a better inference performance of the decision support process, we decide to take a trade-off strategy that criterion 1, 2, 3, 4, 5 are chosen to be translated into ontology. The rest of the criteria will be implemented by external traditional program logic, but the verification result of these criterion will be stored in the ontology knowledge base as well.

3.4 Terminology collection

At first, we tried to utilize some Ontology Learning techniques. After some survey work, Text2Onto[87] was chosen to be the tool that extracts ontology from the criteria document. Unexpectedly, the result of Text2Onto[14] was not satisfactory. After parsing the criteria document in text, only about a dozen of classes were identified, two object properties were

[14]The version we used is here https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/text2onto/text2onto-071109.zip

identified. For the other ontology learning tools, either no download links are provided, or the tool is not runnable. Since automatic extraction of ontology did not work very well. We decided to do it manually.

The first critical task before modeling is to identify the terminology of the ontology. Here a Bottom-Up approach was adopted. According to [88], Bottom-Up approaches start from the most specific concepts and build a structure by generalization; the ontology is built by determining first the low taxonomic level concepts and by generalizing them. This approach is prone to provide tailored and specific ontologies with fine detail grain concepts. We believe that once a terminology is acquired, class definition and class hierarchy will be easily retrieved from the terminology. Then, the definition of object property and data property will correspondingly become easier. In this step, we have utilized card sorting and laddering techniques that are described in [89]. Useful terms were identified and recorded when we roughly browsed the document. In this first step, both nouns and verbs were recorded. Multiple iterations were carried out to make sure we don't miss important terms. Then, we tried to group these terms into different catalogs. For example, "preservative", "fragrance", "stabilizer", "coloring agent", "substance", and "solvent" describe things in the same field, so they should be cataloged into a same group. Next step, we put these grouped terms into "ladders". In other words, terms were organized by "is-a" relationship in hierarchy structure and this structure became the prototype modeling of our ontology. In the previous example, "substance" has a more generic meaning, then it was laddered in a higher level than the others in the hierarchy; the other terms associated it through "is-a" relation in the lower level. At last, a review to all the selected terms were conducted making sure the modeling is complete.

3.5 A modularized modeling

Since we already have a prototype modeling of the ontology composed of the selected terminology. Here in this step, we should translate the modeling into specific ontology syntax.[15] The axioms of class, properties, and individuals should be inserted. Put it more vividly, the output of terminology collection is more like building a skeleton of the ontology; the modeling in this step is more close to enrich the ontology with flesh and blood. As we have stated in the beginning of this paper, a very important issue of our research is reuse. In pursuit of better re-usability, we propose a modularized methodology to separate the entity model (static conceptualization) and rule model (dynamic conceptualization). In other words, we should identify in which part the knowledge about laundry detergent is relatively constant, and in which part frequent changes may take place. As a result of this, in Figure 3.1, we have two kinds of modules: one is the entity module with solid border line, which represents relative static conceptualization; the other is the rule module with dotted border line, which represents more dynamic criterion rules that relay on entity module.

[15] The laundry detergent criteria ontology can be accessed on GitHub: https://github.com/xudaddd/EU-Ecolabel-laundry-detergent-product-criteria-ontology

In our design, still in Figure 3.1, the main module named *laundry_detergent* contains generic concepts, roles and individuals of the domain. For the other more generic entities, module *laundry_detergent* reaches to them via dependencies. In OWL 2 scheme, we can implement this dependency by using *import* syntax, which means an ontology will use all those concepts and relationships from the imported ontology. For our laundry detergent product group, we have entity module *iso_standards*, which contains all the ISO standards references; *ghs_hazard_statement*, in which stores all the hazard statements and codes of GHS (Globally Harmonized System of Classification and Labeling of Chemicals); *regulation_european_commission*, where stores all the European Commission regulation reference; *european_risk_phrases*, where all relevant European risk phrases of chemicals are listed; *commission_decision*, which refers to all relevant European Commission decision documents; *didlist*, which is a database for detergent ingredients. As we have put them into independent modules, they are easier to be imported and reused by other domain ontologies. Please note that, although the main module *laundry_detergent* imports these sub-modules, it does not mean that *laundry_detergent* need all the content in them. Maybe only a part or even a very small part of content is useful for upper level modules.

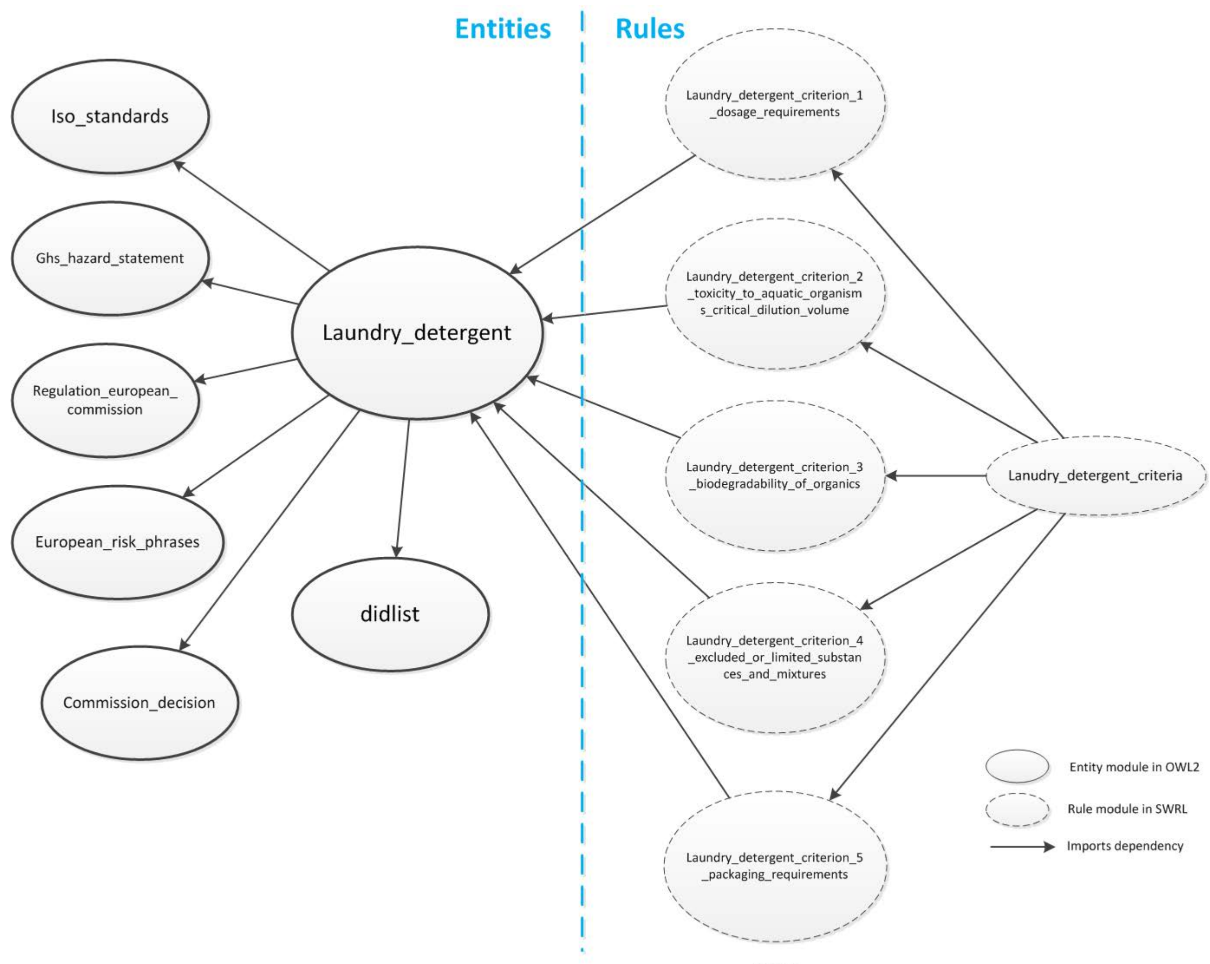


Figure 3.1: Ontology modularization schema for EU Eco-label laundry detergent product group

Module *Laundry_ detergent* This module is the skeleton of the laundry detergent domain ontology. Almost all the important domain concepts and relationship are defined in this ontology module. Figure 3.2, a class diagram in UML illustrates the main classes defined in this module. On the right side of the diagram, we can find a hierarchy of the candidate laundry detergent product and there are five kinds of laundry detergents that are concerned in this criteria: color safe detergent, heavy duty detergent, low duty detergent, fabric softener, and stain remover. Heavy duty detergent and low duty detergent appear more often than the others. The core candidate laundry detergent class is associated with several other parameter classes via object properties. As illustrated in Figure 3.2, each instance of candidate laundry detergent must has at least one kind of chemical as ingredient. It is required to specify the manufacturer of the product, the countries where it will be sold, and the product type. Each candidate laundry detergent should be also associated with one and only one parameter instance for the critical dilution volume, reference dosage, weight utility ratio, aerobically non-biodegradability, and anaerobically non-biodegradability. These parameter classes use various kinds of functional unit e.g. "g/kg wash" means grams per kilo wash. If a candidate laundry detergent doesn't comply to the criteria, it will be cataloged into the rejected detergent class.

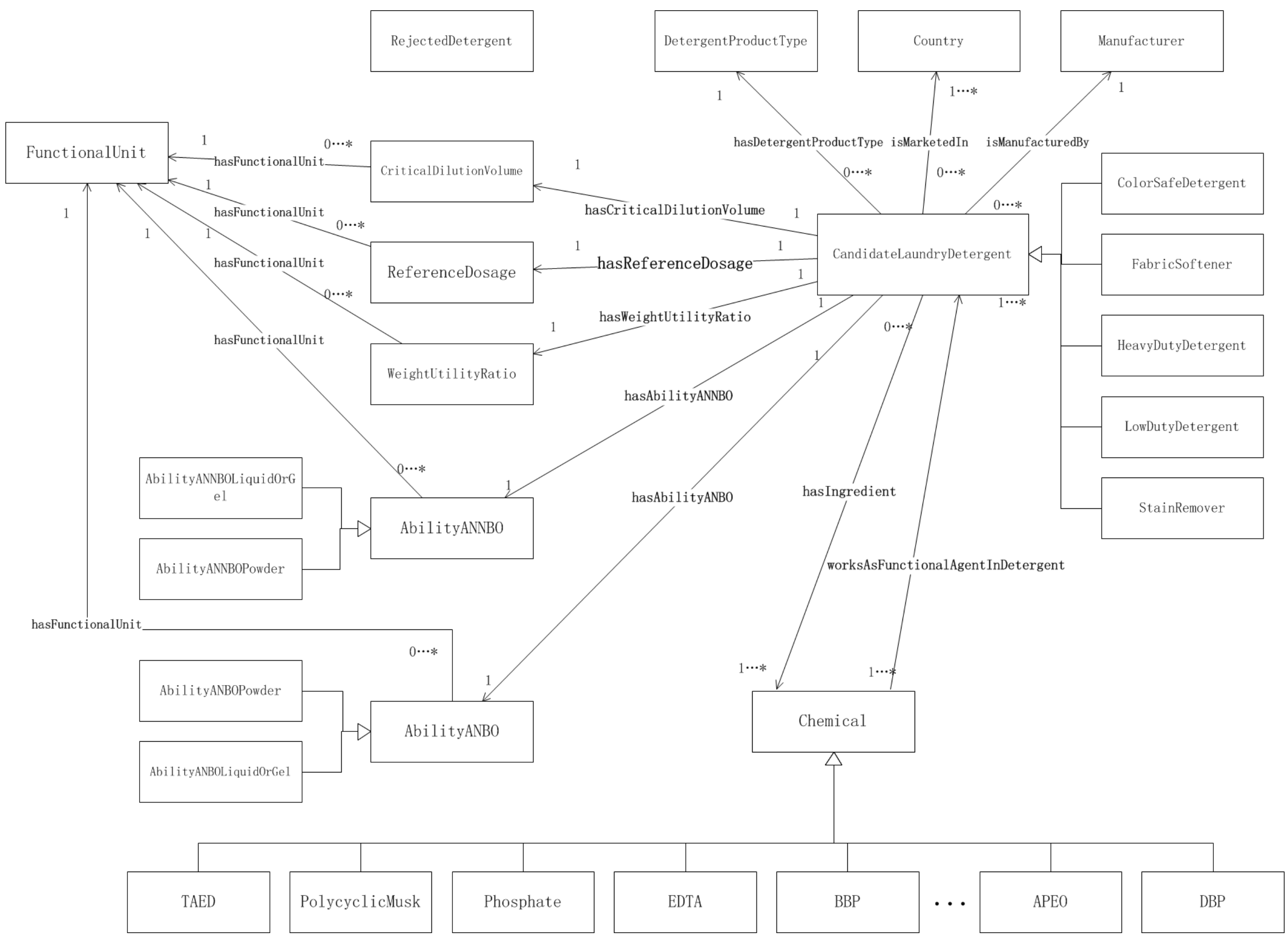


Figure 3.2: Structure of Module *Laundry_ detergent* presented in UML

Module *Didlist* This module is the conceptualization of the detergent ingredient database. In EU Eco-label laundry detergent product criteria, this database is recorded in a excel file, which is not very convenient to be used in applications or other software system. This module is interesting because it will be reused in other product group criteria. We have developed an excel scanner to read this excel file, then generated this module as OWL files. Figure 3.3 is the representation of this module in UML class diagram. In this module, all the detergent ingredients are sub-classified into groups: amphoteric surfactant, anionic surfactant, cationic surfactant, non-ionic surfactant, preservative, and other ingredients. Various functional units are identified by scanning the whole excel file. Full name label and annotation are attached to each of them accordingly. Each ingredient has one and only one anaerobic degradation characteristic e.g. "N" means anaerobically not biodegradable; Each ingredient has one and only one kind of aerobic degradation characteristic e.g. "I" means aerobically inherently biodegradable, but not readily biodegradable.

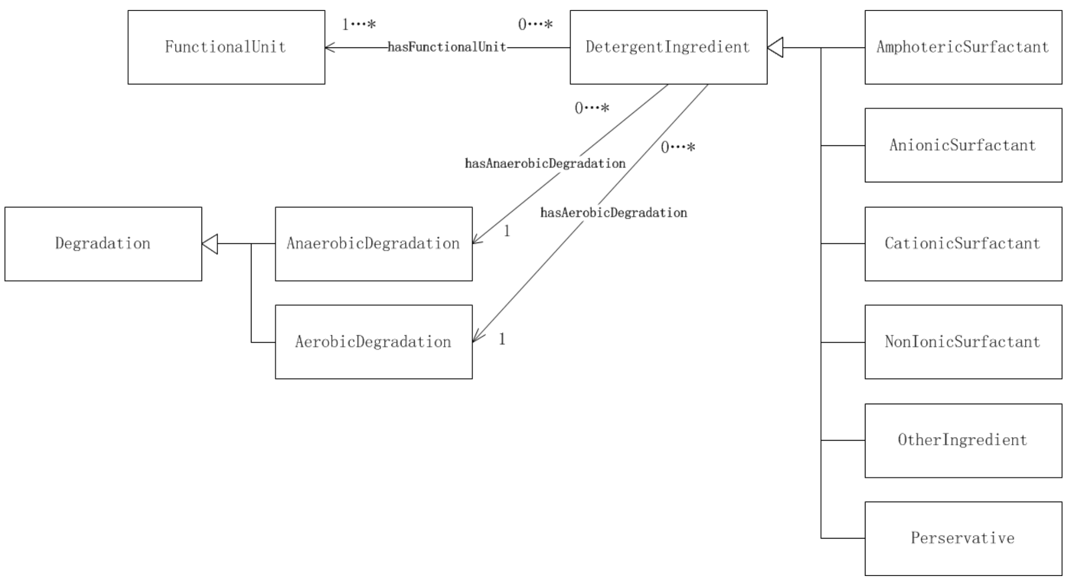


Figure 3.3: Structure of Module *Didlist* presented in UML

Module *European_risk_phrases* This module covers all the European Risk Phrases specification. Since European Risk Phrases is an external standardization reference that appears in criterion 4, it's better to keep these specification to be an independent module. Figure 3.4 illustrates all its classes and individuals. From the picture we can see that, most of this module is the risk phrase individuals. Each risk phrase individual has two data property assertions, e.g. individual "R49" *hasRiskCode* "R49"; *hasPhraseStatement* "may cause cancer by inhalation". This module is reusable in other EU Eco-label product group.

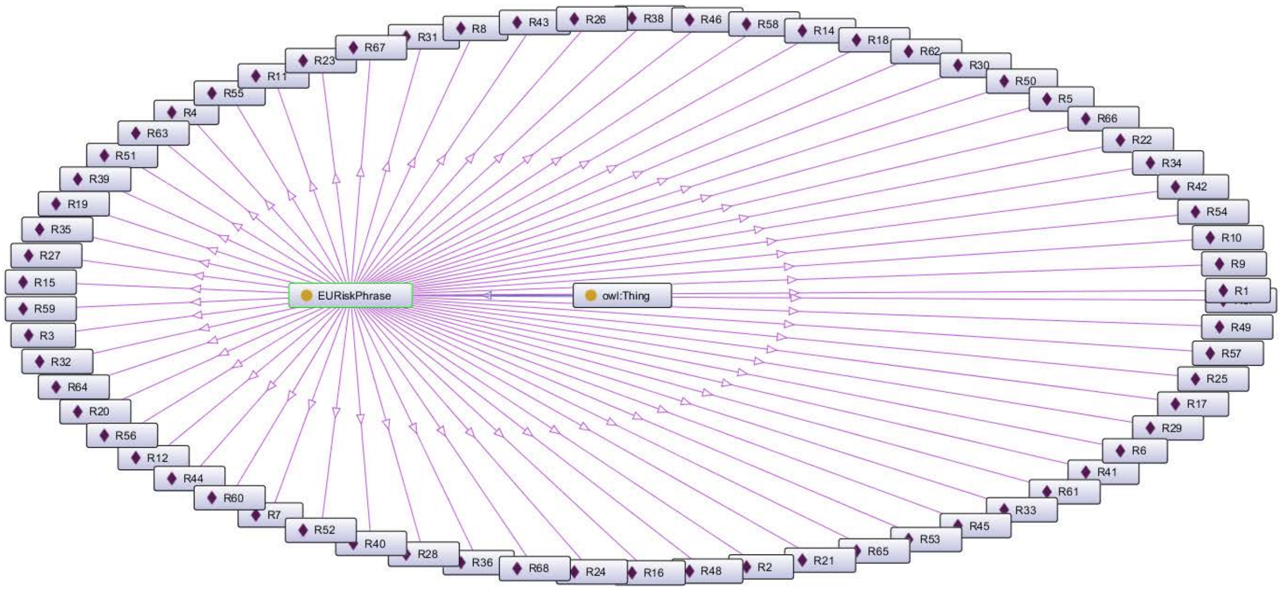


Figure 3.4: Structure of Module *European_risk_phrases* illustrated in OntoGraf Protégé plug-in

Module *Ghs_hazard_statement* Similar to previous module *European_risk_phrases*, GHS (Globally Harmonized System of Classification and Labelling of Chemicals) is also an external reference in criterion 4. A mapping between GHS statement and European Risk Phrases is presented in this criterion. A module following almost the same pattern as module *European_risk_phrases* is modularized. Most of this module is hazard statement individuals. Each hazard statement individual has two data property assertions, e.g. individual "H261" *hasHazardCode* "H261"; *hasHazardStatement* "In contact with water releases flammable gases". This module can be reused in the other EU Eco-label product groups, like all-purpose cleaners, cosmetics products.

Module *Iso_standards*, *Regulation_european_commission*, and *Commission_decision* These modules store the external documentation reference. They record relevant EU documents, standard, commission decision or regulations that are referred in this detergent laundry criteria. These dependency and reference contribute to a better understanding of the criteria in a bigger picture. Figure 3.5 presents the structure of these three modules. Each of these individuals is equipped with URLs that link to external resources. These three modules can be also reused and supplemented by other domains and other EU Eco-label product groups.

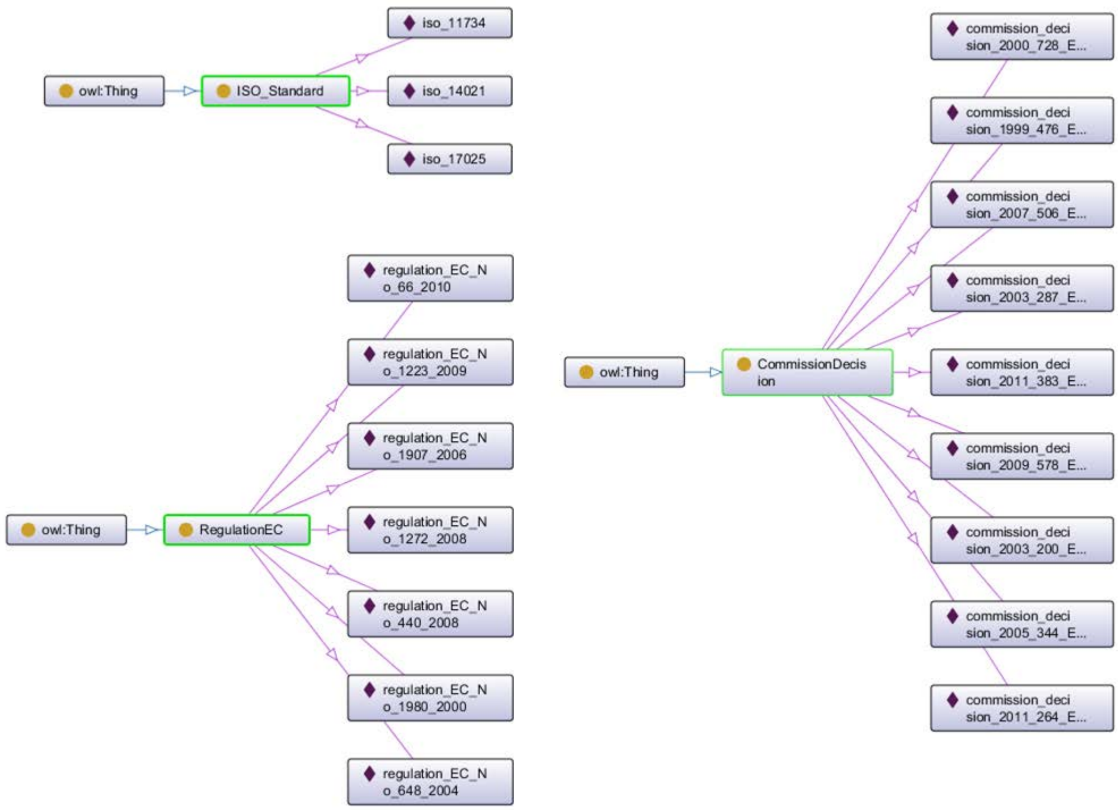


Figure 3.5: Structure of module *Iso_ standards*, *Regulation_ european_ commission* and *Commission_ decision* illustrated in OntoGraf Protégé plug-in

Several advantages exist in this modularized design. As more coherent concepts and relationships are gathered together to form modules, it'll be easier to manage knowledge and data in large scale. Complex conceptualization can be achieved by integrating multiple small modules. Also, it's easier to configure and replace modules rather than to do slight changes directly in a large structure. Take the same example in Figure 3.1. We have a general conceptualization of laundry detergent product which is stored in domain module *laundry_ detergent*. This major ontology module can be replaced by other modules describing other product groups while still making use of sub-modules like *didlist*, *ghs_ hazard_ statement* and *european_ risk_ phrases*, etc. This actually happens in at least two other product groups "rinse-off cosmetic products" and "all-purpose cleaners and sanitary cleaners" which use the same detergent ingredient database (Figure 3.6). Re-usability is achieved by extracting the common knowledge module and have it shared between domains ontologies.

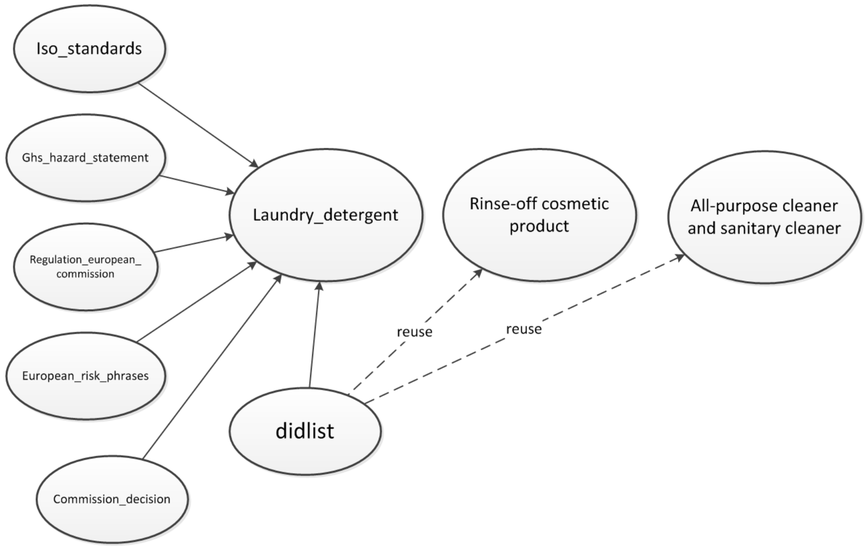


Figure 3.6: Basic reuse pattern which happens between laundry detergents, rinse-off cosmetic products and all-purpose cleaners and sanitary cleaners

Modularization implies separation of conceptualization. In our case, we can see that it will be practical to extract rules from ontology modules. In other words, it's better to keep subjective constraints and world description separated. In [90], it is indicated that a knowledge base is a triple $K = \langle T, R, A \rangle$ formed by a TBox, an RBox and an ABox. RBox represents the rules e.g. SWRL. The idea here is to physically separate RBox from the other two elements. We call this the separation of rules and entities. In the detergent ontology shown in Figure 3.1, ontologies represented in ellipse with solid border are concept-centered, which means the main function of these ontology is to describe the concrete world. These ontologies contains concepts and relationships that are meant to describe or record the facts about real world. On the other hand, as for a product group's guideline or criteria, quite much of these information is involved with human objectives. They are the rules and willingness that human beings impose to the world. In our research, we implement such separation between rules and entities in order to have a better reusability.

For detergent products, the concentration of different chemical ingredients has to comply with certain limit and standard. We can hardly say that such goal-oriented specification is plain description of the world. Moreover, such rules may change time after time. This actually happens, because the product guideline keeps being updated as EU Commission keeps generating new amendments or revise. In our approach, we have each criteria item be an independent module (not completely independent actually, as these rule modules may also have dependencies to other external or internal ontology modules). For example, each of the 5 criterion of the laundry detergent product group is made into an independent OWL file. In the OWL file, firstly, the fundamental entity modules are imported (In Figure 3.1, module hierarchy whose root is *laundry_ detergent* is imported by all the five criterion), then SWRL rule axioms are inserted. As each criterion is distributed in its corresponding module alone, we can easily replace them with new rules and manage them in a configurable way without impacting the others.

At last but not the least, for the criteria ontology as a whole, a single entry module is introduced to include all the criterion, e.g. the *laundry_detergetn_criteria* module on the right side of Figure 3.1. For applications, once the ontology entry is provided, the whole ontology composed of all the entity and rule modules will be retrieved. With this configurable design, expansion and alteration to the ontology will be easier. For example, when a new criterion is about to be approved by the commission, in Figure 3.7, we can update the product criteria to a new version by adding a new rule module and new entry called *Laundry_detergent_criteria_2.0* without losing trace of the previous one. The newly added rule module could be about another new criterion or just an update version of existent criterion. The removal of certain module is similar, all we need is to introduce another entry module. For example, if the new entry module imports criterion 2, 3, 4, 5, and thus criterion 1 will be removed from this version of criteria ontology.

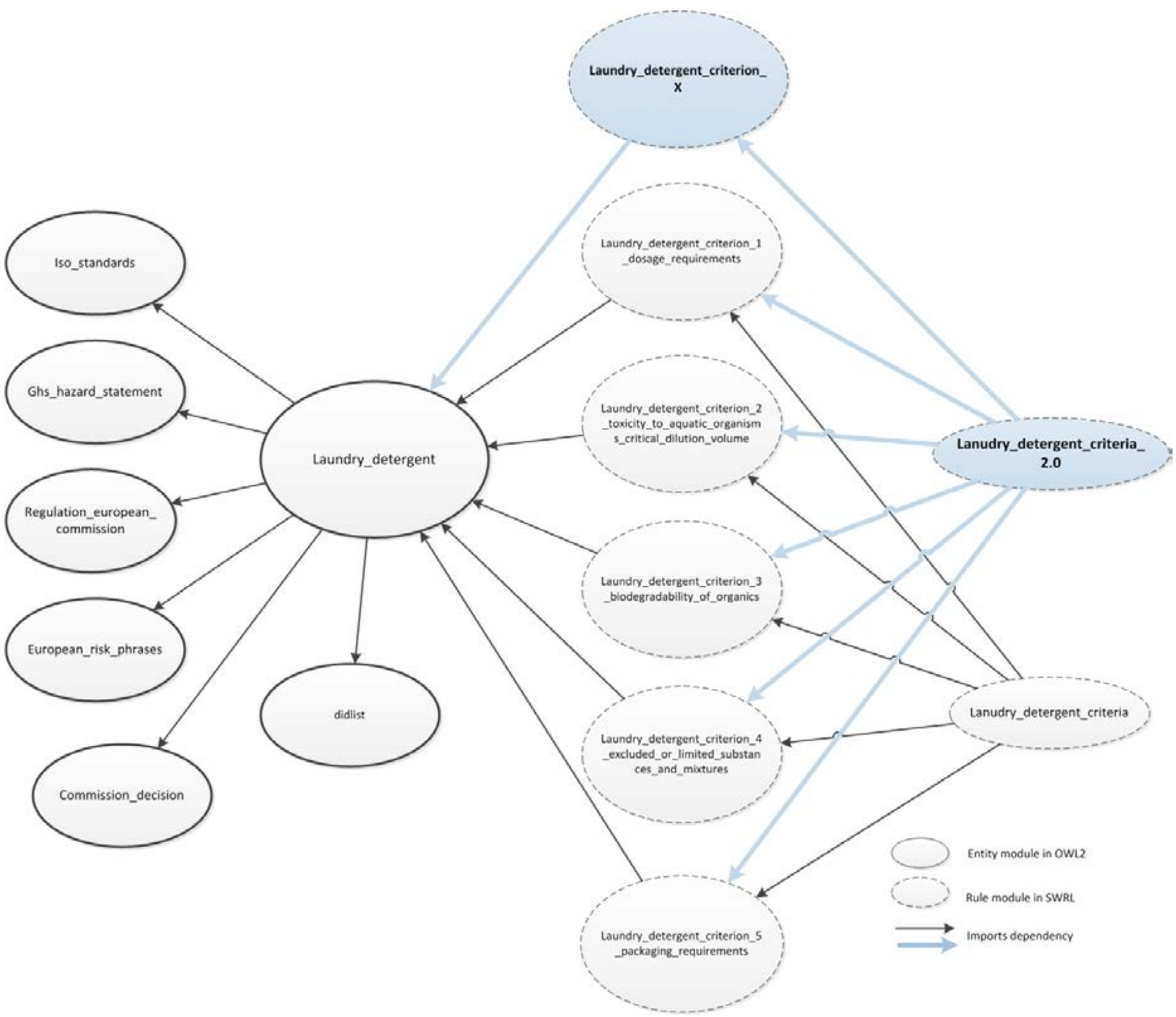


Figure 3.7: Ontology expansion by adding new rule module of criterion for detergent product group

3.6 Evaluation and analysis

The laundry detergent ontology is the first criteria ontology that we have developed for EU Eco-labeling. Another two important criteria ontologies about rinse-off cosmetic product and all-purpose cleaner are under development. All these ontologies will be included in a knowledge base framework. Adjustment and improvement in favor of global performance are being taken into account. Evaluation of single ontology and the whole knowledge base is also undergoing. The advantage of the design of modularization and separation has been observed by researchers as module *Didlist*, module *European_risk_phrases*, etc. can be directly reused by newly developed ontologies.

As we have presented in requirement analysis section, a very important motivation of this ontology development is to judge whether a candidate product is qualified to be labeled. According to a classification of ontology evaluation approaches in [91], our evaluation approach is more close to those based on using the ontology in an application and evaluating the results. We have seen that, this laundry detergent criteria ontology is successfully applied in a decision support system in [92]. Synthetically, taking into account the criteria and aspects introduced in [93] and [94], we have evaluation result as following:

Syntax The criteria ontology is described in standard OWL syntax.

Semantics Since SWRL rules are defined in the ontology and inference support is a basic requirement of our ontology, multiple reasoners e.g. Fact++, Hermit, and Pellet have been applied to check and verify the semantic consistency. So, the ontology is always logical consistent.

Vocabulary Almost all the classes, properties and individuals in the ontology have a meaningful identifier which follows Camel case naming pattern. For those entities that have abbreviation names and vague meaning names e.g. *CDV* and *H400*, a *rdfs:label* axiom is added as complement.

Structure The structure of our ontology is relative simple, the depth of both class hierarchy and property hierarchy is no more than two. The most out-degree for an individual that reaches to other individuals via properties is 14. Taking all the modules into account, 68 classes, 46 object properties, 21 data properties and 460 individuals are defined and stored in our ontology. The number of total axioms is 5786. DL expressivity is *ALCHQ(D)*. Our ontologies can be easily understood and manipulated by other knowledge engineers.

Documentation Each module of the ontology has a textual annotation. For those terms that come from specific domain glossaries, textual annotation and external links are provided. For every SWRL rule, annotation as well as the corresponding anchor position in the document is indicated.

As regards to more specific validation, the competency questions that are defined in requirement analysis section have been translated into SPARQL queries. They work fine with our ontology and correct result can be queried. Here are two examples as listed below and the prefixes we used are declared at the beginning.

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ld: <http://www.enit.fr/xuda/ontologies/laundry_detergent#>

CQ1: If this product is qualified to be eco-labeled?

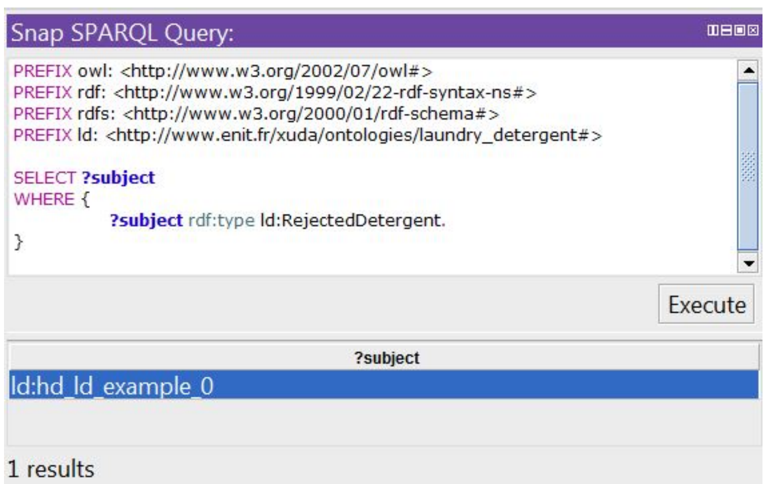


Figure 3.8: Query answer for CQ1, the result shows the ID of product.

CQ2: What is the value of this product's certain physical or chemical characteristics? (Critical dilution volume, biodegradability, weight/utility ratio, etc.)

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ld: <http://www.enit.fr/xuda/ontologies/laundry_detergent#>
SELECT ?subject  ?property ?value
        WHERE {
                ?subject rdf:type ?subType.
                ?subType rdfs:subClassOf ld:CandidateLaundryDetergent.
                ?subject ?property ?object.
                ?object ld:hasValue ?value.
        }
```

subject	property	value
'heavy-duty laundry detergent example NO.0'	hasCriticalDilutionVolume	"30000.0"^^<http://www.w3.org/2001/XMLSchema#double>
'heavy-duty laundry detergent example NO.0'	hasReferenceDosage	"20.0"^^<http://www.w3.org/2001/XMLSchema#double>
test_2_detergent	hasCriticalDilutionVolume	"20000.0"^^<http://www.w3.org/2001/XMLSchema#double>
'heavy-duty laundry detergent example NO.0'	hasAbilityANBOLiquidOrGel	"0.5"^^<http://www.w3.org/2001/XMLSchema#double>
'heavy-duty laundry detergent example NO.0'	hasWeightUtilityRatio	"2.0"^^<http://www.w3.org/2001/XMLSchema#double>
'heavy-duty laundry detergent example NO.0'	hasAbilityANNBOLiquidOrGel	"0.5"^^<http://www.w3.org/2001/XMLSchema#double>

Figure 3.9: Query answer for CQ2.

3.7 Experience and lessons learned

By developing this modularized ontology knowledge base, we have acquired some interesting experience and lessons about ontology design and application. As far as we can see, people have been trying to build more and more complex knowledge representation. If we take documents, which are written in whatever language, as a model or representation of knowledge. To some extent, developing ontology is like a translation process that translates model of human language to formal knowledge representation which can be accessible by machines. As the expressiveness of human language is very high, a computable modeling and translating scheme that has competent expressiveness is needed. The expressiveness and modeling complexity of ontology language has been increasing. We can see this from the evolution of OWL to OWL 2. It is also observed that, in the early days of ontology research, simple knowledge content e.g. medical terminology often used to be the object of study. Today, complex documents e.g. specifications, legal terms, executive orders are expected to be made into ontology. In order to handle more complex knowledge representation or modeling in human language, more comprehensive consideration should be taken into account. The entity-rule separation pattern as well as modularization are such kinds of consideration and exploration that try to handle such more and more sophisticated modeling tasks. As we have discussed in the beginning of Section 4, descriptive entity-related knowledge are relatively constant which means they don't change very much. While, the subjective rule-related knowledge part could be altered frequently. We put them in separation in order to better manage and control the change. The philosophy generalized from this entity-rule separation and modularization pattern can be applied into other modeling or application domain. When dealing with criteria alike knowledge representation, we can apply this entity-rule separation pattern to model descriptive entity-related knowledge and subjective rule-related knowledge into different models, which will facilitate reuse and maintenance.

Figure 3.10 is a more detailed mind map specification for the application of this entity-rule separation pattern. The point of our learned lesson is that before diving into the concrete modeling, necessary higher level abstraction and conceptualization should take precedence. In our case, the target documentation is the eco-labeling criteria. According to the characteristics of the document and the domain knowledge, modularization scheme based on the entity-rule separation pattern is proposed. Then, in each module, the concrete modeling and potential reuse proceed. However, in reality, the boundary of each task could be not very clear. For example, reusability is a very important factor when we decide to set up *Didlist*, *Ghs_hazard_statement* and *European_risk_phrases*. In even more generalized cases and other domains, other modularization schemes are also possible. It depends largely on the objective and application scenario of the modeling. However, in our research, we have seen that, instead of direct and premature modeling, extra work before that is in favor of a good ontology quality and re-usability.

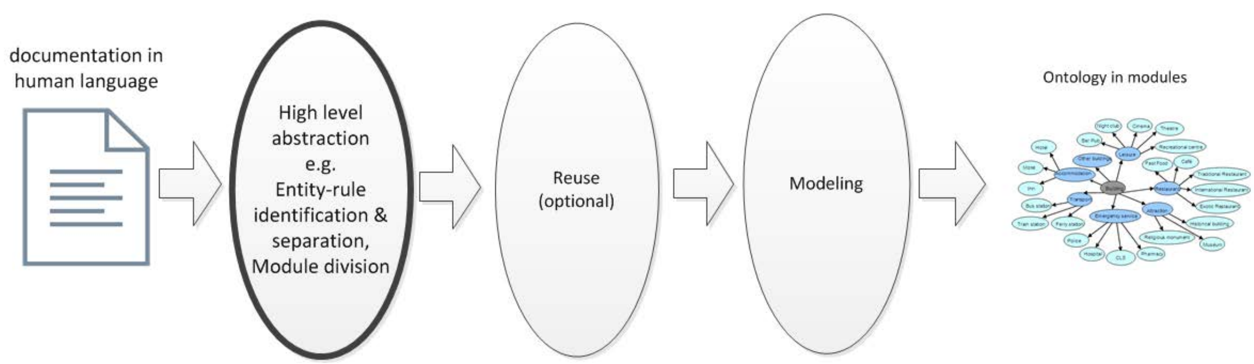


Figure 3.10: Before ontology development and reuse, a high level abstraction e.g. entity-rule separation and module division is often needed.

3.8 Added-value and expectation of the knowledge base

EU Eco-labeling authority Since all the knowledge is retrieved from the criteria documents, this knowledge base will be a good reference for the agents from EU Eco-labeling authority, no matter human expert or computer. The developed ontology will be an important machine exploitable knowledge base to the decision support system for the EU Eco-labeling process. It will be easier to check the relationship between different concepts, check the definition of a concept and query them in various criteria. For example, the knowledge module DIDList (Detergent Ingredient Database List) in our knowledge base records chemical characteristics of hundreds detergent ingredients. It can be reused in other domains' knowledge engineering or application development. Instead of the original excel format, the OWL format ensures a good interoperability with other Semantic Web assets.

Producers and the business For the producer or manufacturer, our knowledge base, especially the ontologies can be also a good reference to check their product's validity to the eco-lebeling criteria, and integrate it into their own environment management system. It can be also an efficient way of product profile exchange and communication with other partners. Take a tourist camping site in EU Eco-label for example, according to the current tourist accommodation criteria, all the cleaning and laundry detergent products used in the site are required to be eco-labeled as well. In this situation, the proprietor of the tourist accommodation site and the down-stream cleaning products producer can use our knowledge base as a consensus for product profile's acquisition and exchange.

In addition, the rules go along with the ontology and the presence of the reasoners allow autonomous pre-check for producers who want their products to be labeled. Once the knowledge base of ontologies is published via some public portal, before the concerned producers start the applying process for EU Eco-label, they can use the ontology together with a reasoning service to do pre-diagnosis and pre-check on their candidate products. We believe that this kind of pre-check service will popularize eco-labeling among producers.

Common users and consumers As for the common consumers, a direct query and search for this ontology knowledge base may be hard. While, a friendly interface may be the solution for the public to have access to eco-labeled products' profile. In the actual running environment of the laundry detergent knowledge base for example, a lot of eco-labeled products' detailed specification will be saved. Exposing product's details to consumers will help the public to understand eco-labeling process, why certain products are eco-labeled and how. Various analysis can be applied to these semantic data and new experience will be learned. Together with these data and analysis result, such a knowledge base based on ontology which can be accessed by the public will help to improve and consolidate the credibility of eco-labeling.

Besides these aspects elaborated above, Figure 3.11 illustrates an eco-system vision that involves a broader eco-labeling industry domain and provides richer applications and services to consumers and enterprises. Our knowledge base can be the foundation of this eco-labeling system or be used in other similar applications and systems.

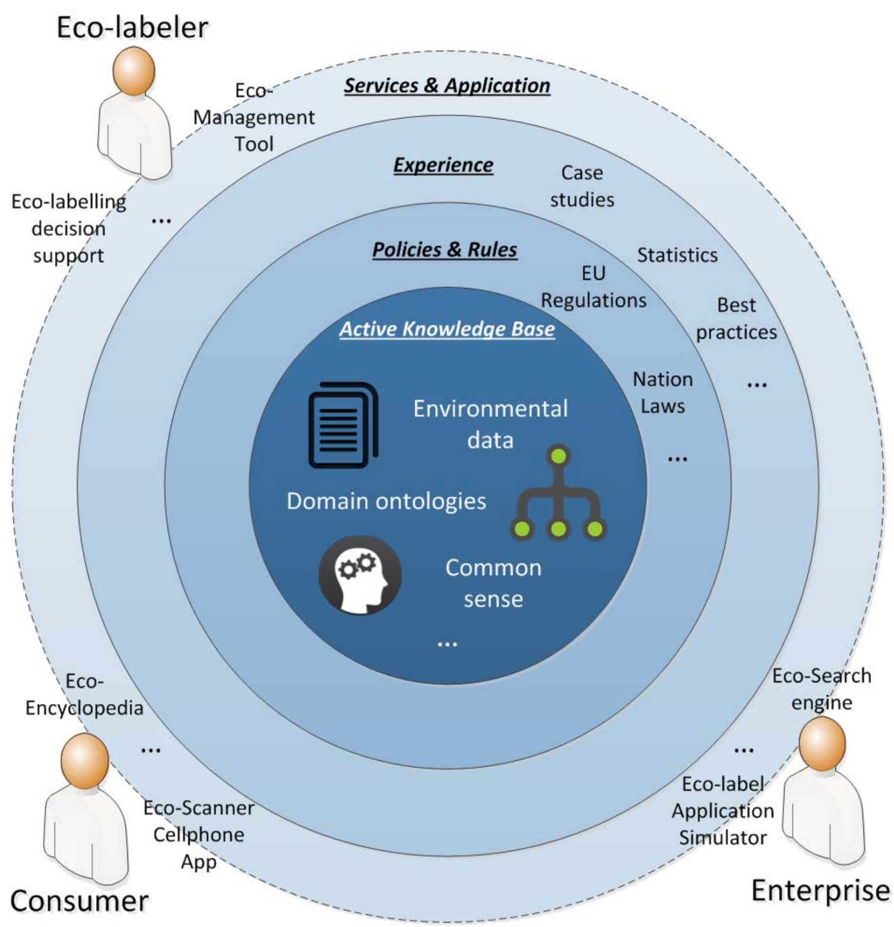


Figure 3.11: A eco-system vision based on large-scale ontology knowledge base and semantic data

In the center of the system, we have an "Active Knowledge Base" constructed by domain ontologies, common sense and raw environmental data. These data can be acquired via various repositories and they don't have to be locally stored. Moving outward to next layer, we have the policies and rules, which contains most of the constraints and regulations for concrete sectors and issues. These regulations or even laws will use the knowledge and data from downstairs layer. For both "Policies & Rules" and "Active Knowledge Base" layers, we should provide a dynamic mechanism that allows modification flexibility and amenity in favor of annex items or deletion. Based upon "Policies & Rules", we have the layer "Experience" containing more advanced information derived from multiple motivation. For example, the best practice may include a commonly accepted and optimized behaviors for certain sectors of business from various aspects which should not be limited within environmental performance. According to the historical data and cases, we can also reach statistics of different criteria. In the same layer, we can archive the failure cases as negative examples. Finally at the top layer, which is also the most interesting one, is the "Service & Application". In this layer, we should easily use some experience or advanced data directly from downstairs layer, or we might as well dig data and knowledge from further down layers. For eco-labeler, various management tools are provided to monitor the labeled products. Based on different priorities, we can expose data to labelers for analysis and accumulate new knowledge and intelligence. For enterprise, we provide the application simulator service to simulate and assess the draft application before the enterprise initiate the real application so as to increase the success possibility. For consumer, we could provide Apps running on smart phones and other personal services. When they do shopping in the market, they can scan the bar code on the product to get detailed information and customized decision support service.

3.9 Conclusion

In this chapter, an OWL ontology knowledge base for laundry detergent is established. We have described our sequential approach to build the ontology from scratch to evaluation. A separation between entity and rules and the modularization of each criterion are proposed to realize better modularity in structure and function. The modularized knowledge base also exposes part of its modules as reference ontologies that could be browsed or reused by other systems in order to achieve data interoperability and knowledge sharing. This is the first contribution of our research. Based on already developed realistic ontology, we can easily proceed to further research on ontology reuse and integration. Also, the decision support system must be built on the basis of existent ontologies. In the next chapter, we will talk about more on modular ontology topics and present an even more flexible ontology reuse and integration scheme called CIMOn (Contextual Integration for Modular Ontology).

Chapter 4

Contextual ontology: a configurable approach to organize modular ontology

4.1 Introduction

In the last chapter, we have presented a domain related ontology building scheme that captures certain eco-labeling related knowledge. However, in a dynamic and inter-connected semantic web environment, single ontology or self-reliant knowledge base is not enough. Take the laundry detergent product that has been previously introduced for example, people would like to establish new product groups in the knowledge base in order to support more different products' eco-labeling service. In this case, instead of developing totally new ontologies, it is better that we can reuse and reorganize existent ontology components. To address such kind of problem, ontology reuse, ontology mapping, and ontology integration technologies are involved. In this chapter, a contextual ontology mapping and integration method in order to realize better ontology reuse and reorganization will be introduced.

4.2 State of art

As we have seen in the state of art chapter, several modular ontology schemes are in favor of ontology reuse. While, more or less, some drawbacks can still be identified from them. [4] listed some problems due to the limitations of OWL and the current, largely undisciplined approach to ontology engineering: Lack of support for localized semantics; Lack of support for partial reuse. Lack of fine-grained organization; Lack of formal support for collaborative ontology building.

Let's first check OWL imports, which is an original mechanism of OWL and dedicated to address ontology reuse and modularity. As specified in the official documentation, when an ontology intends to reuse another ontology in the local circumstance, an XML catalog specification file is the core of the import redirection mechanism. The ontology editor Protégé uses a configuration file called *catalog-v001.xml*, to describe how imports are redirected. This file is like a mapping from ontology name (name IRI and version IRI) to local URI location. Critics about OWL improts are found in [65] where the authors stated that OWL axioms and entities are always global and can affect any entity in either importing or imported modules. In [2], the authors criticize that the semantics of OWL is defined in such a way that all ontologies, imported and importing, share the single global interpretation. As a consequence, imported and importing ontologies cannot describe differently the very same portion of the world, using different perspectives and levels of granularity, without rising a logical contradiction. *Owl:imports* concerns with its ability to import ontologies only as a whole, while it is often the case in practice that only a certain part of imported ontology is of interest to the importing ontology. Furthermore, the OWL imports mechanism is low level and unstructured. For example, OWL has no built-in notion of an "Interface" analogous to the software notion of an Application Programming Interface (API). In [95], the authors talked about a major drawback of current import design in terms of flexibility. This drawback is that the importing ontology is obliged to import all the content of imported ontology modules which means unnecessary axioms could hinder the efficiency of reuse. In the same paper, they try to improve import mechanism by introducing new syntax "ImportModule" that partly import an ontology by means of extracting a ontology module on the basis of an interface signature. Each time an external module is imported, a module extraction or module calculation is necessary to get part of the imported ontology. This seems a good idea to overcome the previously mentioned drawback, but there is no any implementation for this new import syntax yet. (Locality-based module extraction has been implemented in OWL API, but we could not find similar functions in Protégé or similar ontology editor tool.)

When ontology integration take places, some mapping and linking between entities are unavoidable. In [96], the authors propose a composition-based approach for indirectly matching life science ontologies via one or several intermediate ontologies. The basic mechanism of this approach is before doing mapping between two ontologies, each of these two ontologies should be mapped to a third ontology called intermediate ontology. There could be more than one intermediate ontologies between certain pair of ontologies. The mappings to intermediate ontologies can be reused. Intermediate ontologies should have a significant overlap with the ontologies to be matched. When some intermediate ontology becomes big and predominant in the domain, it is called a hub ontology. Because such an ontology can have many mappings to other ontologies and any new ontology can then be aligned with any other by firstly mapping itself to hub ontology. It seems that this approach can only deal with *equivalence* mapping between entities, and intermediate ontology is prerequisite. Also, direct binary mapping is not recommended.

Based on the modularity and reuse philosophy, some researchers proposed scalable approaches to deal with the ontology change and integration. Early in [97], the authors introduced an approach that uses articulations of ontologies to inter-operate among ontologies. In this paper, the individual ontologies will be referred as source ontologies. Articulation rules indicate which terms, individually or in conjunction, are related in the source ontologies. An articulation ontology contains these terms and the relationships between them. The entities or terms coming from different source ontologies are not put together directly, instead, they are integrated by an articulation or linkage between them. The source ontologies are independently maintained and the articulation is the only thing that is physically stored when integration takes place. The articulation ontology commonly uses concepts and structures inherited from individual sources. As such they can be seen as a new knowledge source for upper layers. The authors stated that this approach ensures minimal coupling between the source ontologies, so that the sources can be developed and maintained independently. While, this approach was earlier than the establishment of OWL standard and it accepts ontologies based on IDL specifications and XML-based documents, as well as simple adjacency list representations. There is no support for OWL ontology specification which is already very popular today.

In the work of [71] and [98], a distinction between ontologies and contexts are firstly described. According to the authors, ontologies are shared models of some domain that encode a view which is common to a set of different parties; Contexts are local (where local is intended here to imply not shared) models that encode a party's view of a domain. When an ontology is contextualized or it is called a contextual ontology, its contents are kept local (not shared with other ontologies) and are put in relation with the contents of other ontologies via explicit mappings. We can see that, from certain point of view, a context is actually some circumstance in which several ontologies are linked together on purpose in order to realize certain objectives or functions. A typical scenario of this linkage or integration is ontology reuse. The authors also indicated the need to enrich ontologies with the capability to cope with:

1. The directionality of information flow: we need to keep track of the source and the target ontology of a specific piece of information;
2. Local domains: we need to give up the hypothesis that all ontologies are interpreted in a single global domain;
3. Context mappings: we need to be able to state that two elements (concepts, roles, individuals) of two ontologies, though being (extensionally) different, are contextually related, for instance because they both refer to the same object in the world.

In order to enrich the capabilities above, C-OWL was proposed. New syntax called *Bridge rules* are introduced to represent various relationships between entities that come from different ontologies. (Actually C-OWL is inspired by DDL, and Bridge rules are also defined in the DDL.) A contextual ontology is therefore the pair: OWL ontology, set of C-OWL mappings, where each C-OWL mapping is a set of bridge rules with the same target ontology. The C-OWL mappings are materialized in a XML representation. In our opinion, a significant contribution of this work is the distinction between ontology and context. While, the proposed *Bridge rules* has very limited semantic. Only five kinds of *Bridge rules* (equivalent, disjoint, compatible, more specific and less specific) have been defined. In [4], the authors criticize that DDL has significant limitations with regards to linking of modules with roles. (Since C-OWL is inspired by DDL, this is also the limitation of C-OWL.) For example, roles defined in other ontology modules (i.e. foreign roles) cannot be used to construct new concepts, or to construct new roles from foreign roles.

As we have discussed in the previous chapters, E-Connection [50] is very interesting approach for ontology reuse and multiple linked ontology representation. In their paper, the authors criticize *owl:imports* syntax that: The only way that the *owl:imports* construct provides for using concepts from a different ontology is to bring into the original ontology all the axioms of the imported one. Therefore, the only difference between copying and pasting the imported ontology into the importing one and using an *owl:imports* statement is the fact that with imports both ontologies stay in different files. This certainly provides some syntactic modularity, but not a logical modularity, which would be indeed more desirable. The use of *owl:imports* results in a completely flat ontology, i.e., none of the imported axioms or facts retain their context. While it is possible to track down the originator(s) of some assertions by inspecting the imported ontology, OWL reasoning does not take such context into account. To address these problems and overcome the shortage of *owl:imports*, new syntax called *link property* is introduced to OWL specification. A link property is a binary relation between instances of classes, which belong to different E-Connected ontologies. The source of a link property is the ontology in which it has been declared; the target of the link is the ontology specified in the *owl:foreignOntology* tag in the declaration. However, this *link property* is usually defined in the source ontology, which means extra information is injected into the original ontology and these original ontologies are "contaminated". By using E-Connection and *link property*, we can surely adapt a group of ontologies in a new context, but what if we want to step backwards, take the connection apart and use the same ontologies somewhere else? Also, many restrictions are specified in E-Connection approach. For example a link property cannot be tagged as transitive or symmetric. The ontologies to be connected should be small and disjoint. Thus, a class cannot be declared in an ontology as a subclass of a class declared in a foreign ontology in the combination. A property (object, datatype or link property) can not be declared as sub-relation of a foreign property (it can still be declared as sub-property of a local one); an individual can not be declared as an instance of a foreign class, and a pair of individuals cannot instantiate a foreign property. Similar restrictions and more discussion can be found in [75]. E-Connections also constrain the use of URIs. In OWL-DL, a URI can not be used, for example, both as a class and a datatype, or as an object property and a datatype property. In an E-Connected ontology, a set of additional restrictions must be imposed, namely an URI cannot be used "locally" in two different component ontologies. In [99], there is an application case of E-Connection. Tableau algorithms have been implemented for the E-Connection languages that combine OWL-DL ontologies by either disallowing inverses or number restrictions on the link properties in the OWL reasoner Pellet. More discussion about inference and reasoning implementation on E-Connection can be found in [100]. It is worth emphasizing that E-Connections are not a suitable technique for combining ontologies dealing with highly overlapping domains, which prevents its use in some important Knowledge Engineering applications, such as ontology refinement [50].

To deal with the diversity of ontology languages, ontology modularity and relations, DOL (The Distributed Ontology, Modelling and Specification Language) was proposed in [101]. The goal of this language is to equip heterogeneous ontologies with a precise semantics and proof theory. DOL enjoys the following distinctive features: modular and distributed ontologies are specially supported; ontologies can not only be aligned, but also combined along alignments; logical links between ontologies are supported, etc. An ontology called LoLa (logics and languages) [102] which formally describes DOL's vocabulary for logics, ontology languages (and their serializations) as well as logic translations was developed. This work seems trying to build a formalism or meta model for all ontology languages and mappings, but very few practical aspects are indicated. Very similarly, in [103], an Alignment API was designed and implemented to address ontology alignment representation and manipulation in standard ways. The Alignment API is both an API for representing alignments and for developing, integrating and composing matchers. It comes with a Java implementation. The EDOAL language (Expressive and Declarative Ontology Alignment Language) extends the Alignment format in order to capture more precisely correspondences between heterogeneous ontological entities [104]. From these research work, we can see that, formalizing and standardization of heterogeneous ontology and ontology alignment or mapping are undergoing. However, these efforts provide suggestions only for ontology and ontology mapping themselves. There is a lack of more pragmatic and functional promote on the application of ontology, e.g. ontology reuse and ontology engineering methodology.

In a more practical point of view, it is very interesting to see that [105] proposed an approach that deals with ontology alignment and transformation based on SPARQL. First, an alignment language was developed for expressing complex relations between aligned entities. The alignment format is an extensible format in XML/RDF. Then SPARQL query language is used to transform RDF data according to the alignment. However, current SPARQL specification isn't yet powerful enough for supporting the transformation task with complex mappings. To implement complete alignment framework, the authors proposed two things: first, an implementation of a SPARQL data transformation engine integrating PSPARQL and SPARQL++ and, second, a grounding of an abstract, and expressive alignment language to this new PSPARQL++. This approach could be very practical, because SPARQL is an efficient RDF query language which is already popular and well supported by mainstream programming language. SPARQL extensions for processing alignments will facilitate RDF data integration. While, an obstacle that may hinder this approach is the potential lack of TBox mapping support. Because SPARQL is mostly about RDF graph mapping carried out in ABox. The only sub-language of SPARQL that provides TBox support is SPARQL-DL [106].

In [107], an extension of the DL formalism, IDDL (Integrated Distributed Description Logics) was proposed. This is another Description-Logics-based language that extends standard DL with distributed capabilities. It seems that this research is inspired by DDL and P-DL. The main advantages of this approach, declared by the authors, are separation of local semantics (which is standard DL), and global semantics; it allows composition of ontology mappings. Because of the separation of semantics, an IDDL knowledge base contains two components: a family of local DL ontologies and a family of ontology alignments. The distributed system's or the knowledge base's semantics depends on local semantics, but does not interfere with it. In our opinion, this independence or semantics separation keeps local semantics intact, and that is vital to modularity and reusability. We can also see that IDDL owns a richer mapping semantics as 6 possible types of correspondences between ontologies are defined in IDDL, while only 4 bridge rules are defined in DDL. As for the inference, correspondences and axioms from several ontologies are used to deduce new axioms or correspondences. The only shortcoming of this research work may be the lack of examples and application cases. Although the authors have stated that this approach still needs theoretical investigation, it already inspired our work a lot. We have presented a table (Table 2.2) comparing several available modular ontology methods at the end of Chapter 2 State of the Art. More discussion about modular ontology can be found in the Modular Ontology sub-section in Chapter 2.

4.3 Contextual Integration for Modular Ontology method

In this part, we will present our Contextual Integration for Modular Ontology method and discuss the problems that we have described in previous sections. According to [71], an ontology is contextualized, or that it is a contextual ontology, when its contents are kept local (and therefore not shared with other ontologies) and are put in relation with the contents of other ontologies via explicit mappings. Contexts encode not shared interpretation schemes of individuals or groups of individuals. Contexts are easier to define and to maintain. They can be constructed with no consensus with the other parties, or only with the limited consensus which makes it possible to achieve the desired level of communication and only with the "relevant" parties. On the weak side, since contexts are local to parties, communication can be achieved only by constructing explicit mappings among the elements of the contexts of the involved parties; and extending the communication to new topics and/or new parties requires the explicit definition of new mappings.

Let us think about the scenario in Figure 4.1. Imagine that we have two ontology modules as represented by the rounded rectangles. In $Module1$, we have two classes A and B which are related by property x. In the other module we have a class C. $Module1$ and $Module2$ come from different domains respectively. Suppose that we run into a situation where both modules are needed. Actually, this situation might be very often because in today's web, where considerable numbers of ontologies are developed. Ontology reuse and integration is becoming a convention in knowledge representation and knowledge engineering. Cross-domain ontology composition is not a rare thing. In such a situation, possibly, new relationship is needed to facilitate the composition, for example the relation (property) $r1$ that associates Class A and Class C which are from different modules. (By the syntax of description logics, we have naming pattern for entities that concept corresponds to class, role corresponds to property and instance corresponds to individual. While, it seems that even in DL context, very frequently, individual is still used.) Here we do not specify the domain or range of this property x for that it does not really matter. Also, we ignore the whole interpretation i.e. ABox for now. Instead, we will demonstrate how context works in terms of TBox.

As we have introduced in the state of art chapter and sections, there are already some solutions for this ontology integration or composition issue. OWL, the language itself provides an importing mechanism that supports basic ontology module integration and composition. To deal with the situation in Figure 4.1, if we apply original import syntax of OWL, we should let one module be imported by the other. Then the new relation $r1$ shall be defined in either the importing module or imported module. Two risks can be found here. First, the importing module becomes dependent on the other one not only in this integration scenario. Since import syntax is injected in the ontology module, hence, any other ontology module that wishes to reuse or import the importing module, it has to consequently import all the ontologies that the module has already imported. Second, the placement of new relation $r1$ will inevitably affect the semantics of original ontology modules. In other words, the new semantics of new relationship $r1$ might contaminate original ontologies. A practical but not economic solution for this situation is the duplication of ontologies. Take the same example of Figure 4.1 in the same ontology composition or integration scenario, we would have $Module1$ import $Module2$, apply new relation $r1$ in $Module1$, as illustrated in Figure 4.2. At the same time, a copy of $Module1$ is reserved for other composition or reuse. In our practice of ontology development and research, this duplication approach actually works, while, it may result in redundancy problem. Once original ontology is modified, all the duplication of this original ontology has to be updated. Now, we have seen the problems of ontology importing mechanism, and that is why we think about separating original ontologies from ontology integration context.

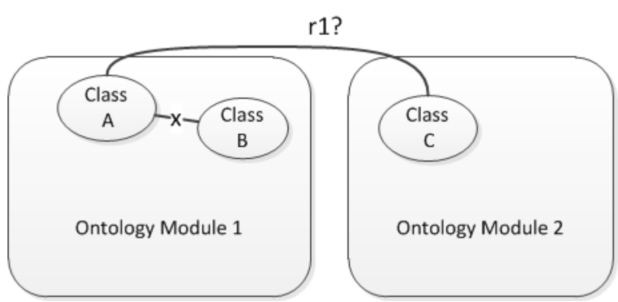


Figure 4.1: In case of ontology reuse or integration, cross module relation is common.

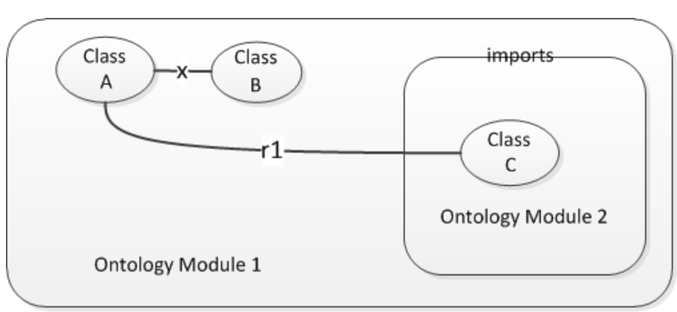


Figure 4.2: A straightforward solution is to let one module "swallow" the other, and introduce new relation in either. e.g. *owl:imports*.

It is necessary to point out that, many ontology composition or distributed ontology schemes try to stress the completeness and logic consistency of the integration. However, they seem to have overlooked problems we have introduced above. To our knowledge, for example, E-Connection and P-DL specify "non-neutral" mapping syntax that is attached to original ontology modules. One of the contribution of our work is trying to stress the importance of intact original ontology modules and the separation between ontology and ontology integration context.

To avoid the risk and problems introduced above, we propose CIMOn method (Contextual Integration for Modular Ontology method). As illustrated in Figure 4.3, when two ontology modules are about to be integrated, a neutral *Context* will be used. Instead of putting extra relation e.g. $r1$ into either original ontology module, we will let this context module store such information and configure the integration. In order to have valid targeting subject and object for added relation, equivalent position takers must be defined in the context module. For example in Figure 4.3, position takers of $ClassA$ and $ClassC$ are defined as $ClassA'$ and $ClassC'$. Notice that position taker is not obliged to contain full definition of the target entity that it represents. Actually, it's more like a pointer that identify the real entity in original ontology module. We might as well call this pointer as Equivalence Corresponding (EC). In reality and actual implementation, entity's IRI is used to label the position taker. As we have defined the substitution of entities in the context module, extra integrating relations will be defined between these position takers. Compared to Figure 4.2, the scheme we have presented in Figure 4.3 has implemented the separation of contextual information and the original ontologies. The new relation $r1$ is in neither ontology modules. Thus, these integrated ontology modules will not be "contaminated" and can be directly reused by others.

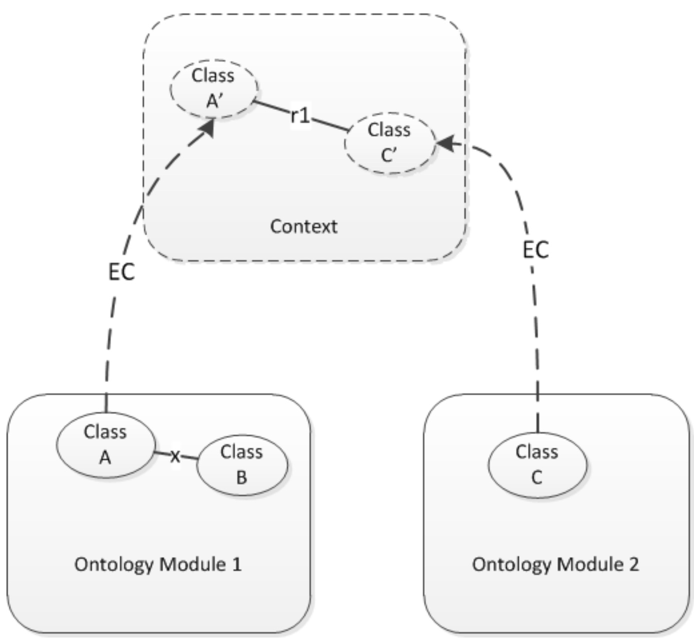


Figure 4.3: Original ontology modules are kept intact by introducing independent context.

Now, we give formal definition of the Context and Contextual Integration.

Context: Given a set of ontology $O = \{O_i\}_{i \in I}$, each O_i is expressed in Language L; S_i is the selected subset of the signature of each module O_i, i.e. $S_i \subseteq \mathrm{Sig}(O_i)$, $S = \{S_i\}_{i \in I}$ is the collection of all the selected signature of each modules; Let r be an arbitrary relation expressed in L, r_{ij} associates two entities from $Sig(O_i)$ to $Sig(O_j)$, $R = \{r_{ij}\}_{i \neq j \in I}$ is the collection of such relations that happen between ontology pairs $\langle O_i, O_j \rangle_{i \neq j \in I}$. Then, for ontology set O and the selected signature collection S, we have $Context_O^S = R$.

Contextual Integration: Based on the definition of Context, let $M_i = O_i^{S_i}$ be the module extracted from O_i by signature S_i, $M_O^S = \{M_i\}_{i \in I}$, then the union of M_O^S and $Context_O^S$ is called the Contextual Integration of ontology set O based on signature collection S. i.e. $CI_O^S = \{M_O^S, Context_O^S\}$.

In more general ontology integration or composition scenario like Figure 4.4, it is not hard to see that one ontology module is reused multiple times. For instance $ClassC$ which is defined in $Module2$, this time it is referred in both $Context1$ and $Context2$. In a different context, the use of the same entity could be totally different, e.g. $ClassC$ could be the domain of property $r1$ in $Context1$, on the other hand, $ClassC$ becomes targeting range of property $r3$ in $Context2$. This polymorphism allows flexible contextual configuration. A big advantage of this scheme is that it allows using the same group of ontology modules to construct various ontology integration or composition without ontology duplication and redundancy.

Prerequisites should be noticed that, in our research, an overlapping-oriented strategy is utilized. According to [63], "in a disjointedness-oriented strategy, one may simply create as many copies of a concept as needed to allocate one copy to each candidate module, and then forget about the duplication, i.e. consider each copy as a separate piece of knowledge, independent from the other copies of the same concept. In an overlapping-oriented strategy, concepts may be directly allocated to multiple modules (without being duplicated) and the system keeps awareness of this multiplicity. Using this awareness the system can let users navigate from one module to another, i.e. from one instance of a concept in one module to another instance of the same concept in another module." Also, the newly formed contextual integration should be guaranteed to be an conservative extension [99] [108] [109] [110] [75] [63] of integrated modules, even though ontology modules are not forcefully disjoint and arbitrary relation could be introduced. In particular, the main intuition behind conservative extension is to ensure local completeness of the modules such that the knowledge contained in each individual module will not be altered even after their integration. That is, integrating modules should not induce new relationships between existing concepts in any existing module. More formally, for two ontologies $O' \subseteq O$ and a signature Σ, O is a deductive Σ-conservative extension of O', if for every axiom α with $\Sigma(\alpha) \subseteq \Sigma$ we have $O \vDash \alpha$ if and only if $O' \vDash \alpha$ [75]. In other words, robustness under replacement [95] [63] should be kept when the context is being edited.

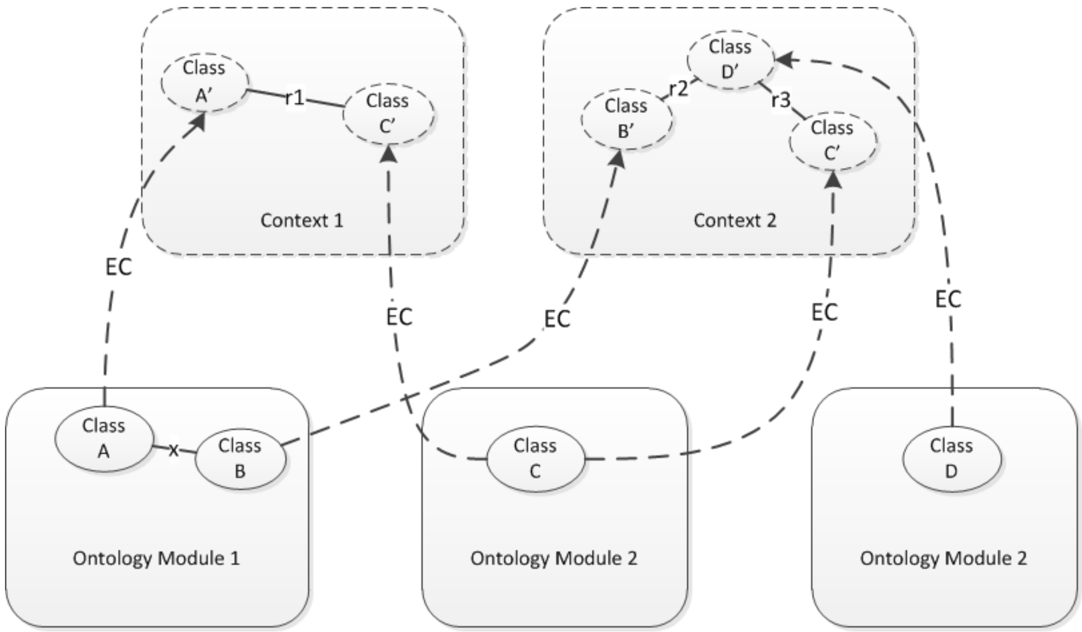


Figure 4.4: By using context, multiple reuse at the same time without duplication is possible. Class C for example, is reused two times in both Context 1 and 2.

According to the official documentation of OWL 2 [12], A OWL API has already implemented the importing mechanism and a *catalog.xml* file will be generated by ontology editor Protégé. In our design, import mechanism should not be discarded. Instead, CIMOn will be compatible with this existent import mechanism. This compatibility of import means that if some ontology module is included in the context, then all the sub-ontologies that this module imports will be included in the context as well. In other words, we regard an import closure as the basic structure to be integrated in the context.

4.4 Layered knowledge base architecture

In the last section, we have introduced CIMOn method that realizes ontology integration and composition. Now we will have a review of this scheme from a higher level and demonstrate how to apply this scheme to ontology engineering practice.

Figure 4.5 is an extension of Figure 4.4. As we can see that three layers are identified in the complete architecture of CIMOn. In the bottom ontology layer, we have a repository of ontologies. All kinds of ontologies coming from different domains will be maintained. Above ontology layer lies the context layer. Just like what we have introduced in the last section, different contexts will be configured here to materialize different ontology integration or composition. For each configured context, it is also like a "Virtual Ontology" that assemble

[1] http://protegewiki.stanford.edu/wiki/Importing_Ontologies_in_P41
[2] http://protegewiki.stanford.edu/wiki/How_Owl_2.0_Imports_Work

several modules. The very top layer is application layer where concrete functions and business are implemented. The context layer is like a middle ware that unites application and ontology. Applications focus on their business functionality, they will not be deeply involved into ontology management and maintenance. Context layer and ontology layer together constitute the ontology knowledge base.

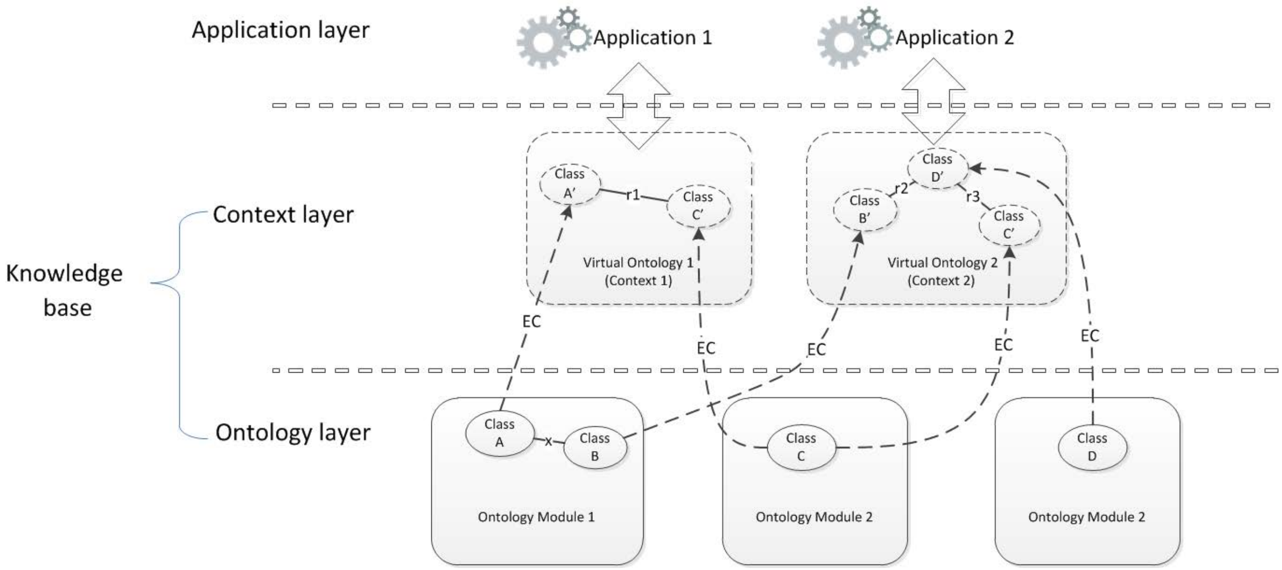


Figure 4.5: CIMOn layered architecture to support knowledge base and application.

Actually, similar design of this contextual ontology can be found in many other frameworks. We will take two examples to better present the idea of this design. The first example is the view in modern database management system. Unlike ordinary tables in a relational database, a view does not form part of the physical schema: as a result set, it is a virtual table computed or collated dynamically from data in the database when access to that view is requested.[3] In our design, the configured context is somehow like the view of tables in database. Another example is Maven framework which is a build automation tool used primarily for Java projects. Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository, and stores them in a local cache.[4] Usually, each project that uses Maven needs a POM(Project Object Model) file to configure their dependencies to libraries and plug-ins. This POM is a bit of like our context configuration for that each application will reach to the ontologies that it needs via this context. The ontology modules that an application needs is just like the dependencies to Java libraries that a Java project has.

In practice, applications in the upper layer needs a configuration context to access those ontology modules in the bottom layer. Obviously, potential advantage of this architecture is that several contexts can be assigned to the same application. If an application needs to

[3]https://en.wikipedia.org/wiki/View_(SQL)
[4]https://en.wikipedia.org/wiki/Apache_Maven

update its ontology knowledge base, it needs to reconfigure its context or change to another new context. Or, the application can choose to return to previous context of ontologies. In order to manage the connection between application and context configuration, a service program has to be implemented in the context layer. In this service program, a set of API will have to be implemented to support manipulation and control from the upper application layer.

4.5 An XML implementation of CIMOn

Figure 4.6 is the XML implementation example of context configuration. The reason why we choose XML as the configuration format is that XML has been widely used and well developed. A lot of API in various programing languages support XML. Also, XML can be easily transmitted via the web. The root element of this configuration file is *context*. Key elements for a *context* are *contextName*, *contextMembers*, *contextRoot*, *contextFilter*, and *binaryMappingBundle*. The appearing sequence of elements won't change the meaning of the context. These elements are listed and explained as the following:

```xml
  <?xml version="1.0" encoding="UTF-8"?>
- <context>
    <contextName IRI="http://www.enit.fr/xuda/ontologies/test_context_0">test_context </contextName>
  - <contextMembers>
      <owlOntology IRI="http://www.enit.fr/xuda/ontologies/iso_standards"
        localPath="C:/Users/pbook/Desktop/Doctor/EclipseWorkspace/ECOLABEL_KNOWLEDGEBASE/test_context/test_iso_standards.owl"/>
      <owlOntology IRI="http://www.enit.fr/xuda/ontologies/regulation_european_commission"
        localPath="C:/Users/pbook/Desktop/Doctor/EclipseWorkspace/ECOLABEL_KNOWLEDGEBASE/test_context/test_regulation_european_commission.owl"/>
    </contextMembers>
    <contextRoot IRI="http://www.enit.fr/xuda/ontologies/regulation_european_commission"/>
  - <contextFilter>
    - <contextMemberFilter IRI="http://www.enit.fr/xuda/ontologies/iso_standards">
        <classFilter mode="selectAll"/>
        <objectPropertyFilter mode="selectAll"/>
        <dataPropertyFilter mode="selectAll"/>
        <individualFilter mode="selectAll"/>
      </contextMemberFilter>
    - <contextMemberFilter IRI="http://www.enit.fr/xuda/ontologies/regulation_european_commission">
        <classFilter mode="selectAll"/>
        <objectPropertyFilter mode="selectAll"/>
        <dataPropertyFilter mode="selectAll"/>
      - <individualFilter mode="part">
          <filterItem reserveOrNot="false" entityIRI="http://www.enit.fr/xuda/ontologies/regulation_european_commission#regulation_EC_No_1272_2008"/>
          <filterItem reserveOrNot="true" entityIRI="http://www.enit.fr/xuda/ontologies/regulation_european_commission#regulation_EC_No_66_2010"/>
          <filterItem reserveOrNot="false" entityIRI="http://www.enit.fr/xuda/ontologies/regulation_european_commission#regulation_EC_No_1907_2006"/>
        </individualFilter>
      </contextMemberFilter>
    </contextFilter>
  - <binaryMappingBundle targetOntologyIRI="http://www.enit.fr/xuda/ontologies/iso_standards" sourceOntologyIRI="http://www.enit.fr/xuda/ontologies/regulation_european_commission"
    mappingName="test_binary_mapping_bundle_0">
    - <binaryMapping type="OWL/XML">
      - <SubClassOf>
          <Class IRI="http://www.enit.fr/xuda/ontologies/regulation_european_commission#RegulationEC"/>
          <Class IRI="http://www.enit.fr/xuda/ontologies/iso_standards#ISO_Standard"/>
        </SubClassOf>
      </binaryMapping>
    </binaryMappingBundle>
  </context>
```

Figure 4.6: XML implementation example of context configuration.

1. **contextName** The unique ID for current ontology context. The value of this element is a string name customized by user. Also, an IRI attribute is allocated so that the context can be indexed on Semantic Web. Thus, both the string name and the IRI make up the unique ID of current context.

2. **contextMembers** This element will specify all the member ontologies that are about to be integrated. The IRI of the ontology is indicated. If the ontology is locally stored, a local path attribute is indicated too.

3. **contextRoot** We have observed and learned that, once several ontologies are integrated, the directionality of mapping or inter-relation will affect inference result. A very simple scenario is like this, ontology O_A is related to ontology O_B via a bunch of relations, then the reasoning or inference result of taking into account axioms of O_A at first, then taking into account O_B's axioms could be different from the opposite way. In other words, from a practical point view, suppose that reasoner is launched in ontology O_A, then we have an inferred ontology $R(O_A)$ (i.e. the reasoning result of O_A, here we assume that the reasoning calculation is like a function and we use $R()$ to represent the reasoning). $R(O_A)$ may contains new inferred axioms from A. Then we have all the axioms from O_B applied in $R(O_A)$, for the second time, the reasoner is launched, then we will get a reasoning result of the set of $R(O_A) \cap O_B$. We call the result of this second time reasoning $R(O_A \rightarrow O_B)$. The point here is that, according to different research [111] [112] [71] [50] [98] that have been done for now, it's observed that $R(O_A \rightarrow O_B)$ is not always equal to $R(O_B \rightarrow O_A)$. In more generic cases, the sequence that is used as the parameter of reasoning function R() is not only liner sequence, it might be a tree or even a directed graph. In Chapter 10 of [113], the structure that represents dependency between modules is called dependency graph. We agree and have adopted the two major rules for the dependency graph that: First, there should not be any cycle in the dependency graph of the resulting modularization; Second, If a module reuses another one, it should not directly or indirectly reuse a module on which the reused one is dependent. These two major rules will make sure that there is no cycle neither transitive dependencies in the dependency graph, thus a safe and complete "reasoning grow" will be ensured. Imagine that we have dozens or even more ontology modules to be integrated, one or several pivotal modules are selected as the baseline of the new ontology composition artifact, as the composition grows through *binaryMappingBundle*s, a dependency graph of tree or more complex directed graph will appear. We will maintain this dependency graph when a context is built. Here the element of *contextRoot* works as the beginning of such a growing path of tree or forest. One or multiple IRI attributes will be indicated in this element. They are very important for the reasoning of the ontologies integrated in context. Figure 4.7 shows a typical topology example of the integrated ontology modules (OM) in a context. Every time a context is formed, such a topology that indicates reasoning dependencies between ontologies must be specified by user. This topology will be very important for an incremental reasoning, because by controlling the grow of the tree that starts from root module, sub-tree or sub-graph of the context will be captured. Reasoning and validation can be applied to these sub-division of the context in order to locate any unexpected logic inconsistency or error.

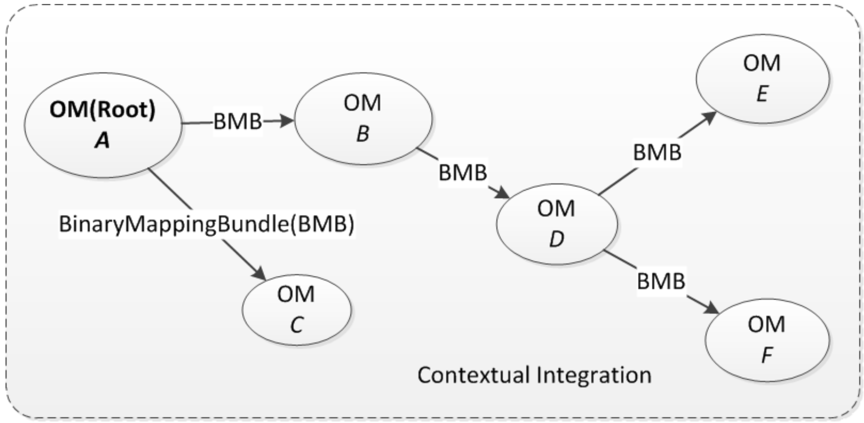


Figure 4.7: Topology example of the integrated ontology modules in a context.

4. **contextFilter** This element is the check-list for all the needed entities in member ontologies for the context, i.e. signature for each integrated module. Meanwhile, it will filter out the unnecessary ones. For now, the atomic filtering granularity is entity level, i.e. class, object property, data property, and individual. Axiom level filtering which is of finer granularity than entity has not been considered and implemented, we will talk about this in the discussion and future work chapter. Each context member ontology will be mapped to a *contextMemberFilter* element. In each *contextMemberFilter* element, 4 concrete filters are indicated corresponding to the 4 types of basic entities that an ontology has. For each *entityFilter* element(*classFilter*, *objectPropertyFilter*, *dataPropertyFilter*, and *individualFilter*), 3 modes are defined: *selectAll*, all the entities shall be reserved; *deselectAll*, none of the entities will be reserved; *part*, entities are partly reserved. If the mode for an entity filter is *part*, then detailed *filterItem* will be offered to further indicate which entities should be reserved and which ones should be discarded. For each *filterItem*, IRI attribute is provided to identify the entity in its ontology.

   It is stated in [114] that it is especially important to ensure that a module extracted from an OWL ontology for reuse or maintenance purposes preserves the results of reasoning tasks. In other words, if we are about to reuse a concept named A and retrieve a fragment T' of the original ontology T, we want to make sure that A, as well as all its sub-concepts, super-concepts and instances are included in T'. In our research, an user's entity granularity selection is provided. Logical completeness is sacrificed for more flexibility and usability.

5. **binaryMappingBundle** This element is used to represent a bunch of inter-relation that takes place between two ontologies. It is easy to see that multiple relations will take place between any two ontology modules in a context. We can call this set of relations between any two ontologies *binaryMappingBundle*. In order to indicate the reasoning directionality, source ontology IRI attribute and target ontology IRI attribute are equipped. Figure 4.8 shows the composition of binaryMappingBundles. In the XML configuration file, each relation that is bundled is contained in an element called *binaryMapping*. Notice that "mapping" may not be a proper name for that not only equiva-

lent mapping relation can be defined. Various relation expressions can be found in *binaryMappingBundle*. As we know, despite the fact that OWL has become a popular standard for representing ontology, there is still no agreement or popular standard for defining ontology mappings between ontologies. Basic OWL 2 modeling expression elements such as "SubClassOf", "ClassAssertion", "EquivalentClasses", "DisjointClasses", "ObjectPropertyAssertion", "NegativeObjectPropertyAssertion", "SubObjectPropertyOf", "ObjectPropertyDomain", "ObjectPropertyRange", "DifferentIndividuals", "SameIndividual", "DataPropertyAssertion", "NegativeDataPropertyAssertion", "DataPropertyDomain", "DataPropertyRange"[5] have been implemented in OWL/XML syntax in *binaryMappingBundle*. Among these expression elements, an even more broad semantic expandability is provided to "ObjectPropertyAssertion" for that not only the object properties declared in the integrated ontologies can be used in this relation expression, but also third-party ontologies' object properties can be used, e.g. OBO Relations Ontology [6].

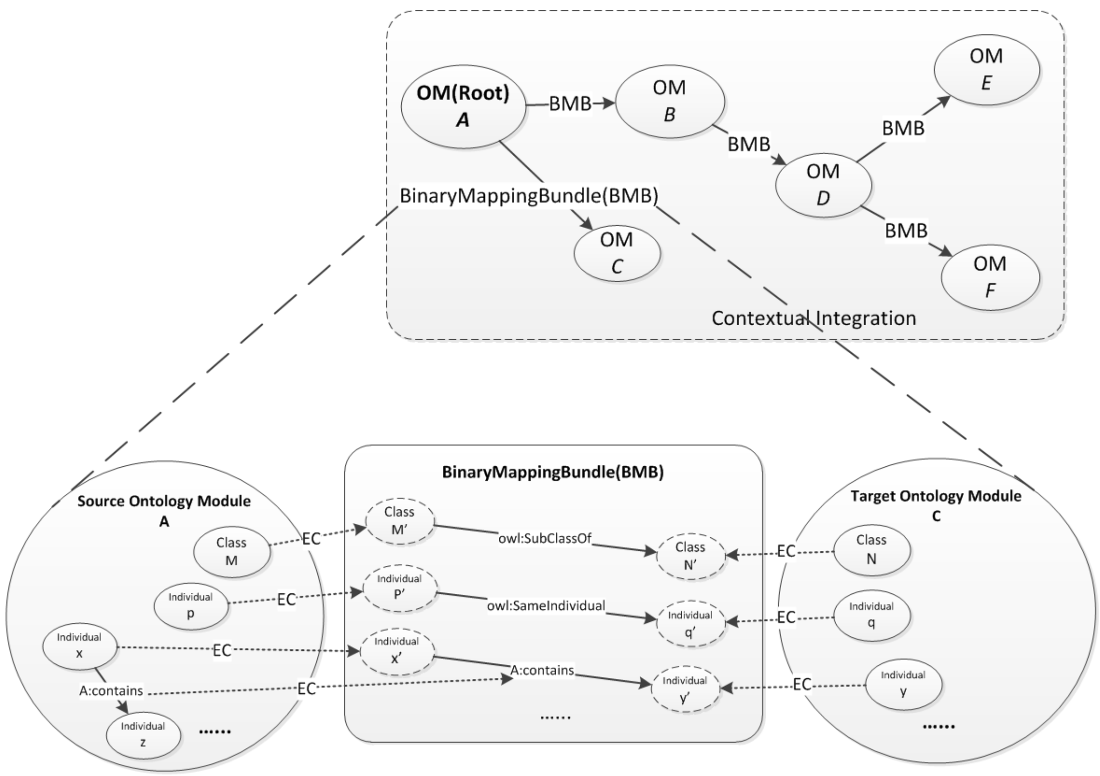


Figure 4.8: A binaryMappingBundle is a collection of all the relations that take place between two modules in context.

A detailed specification of context configuration has been presented above. With CIMOn and its XML implementation, we can realize the layered knowledge base architecture which has better flexibility and re-usability as described above. More details about the concrete

[5] https://www.w3.org/TR/owl2-primer/#Domain_and_Range_Restrictions
[6] http://obofoundry.org/ontology/ro.html

implementation and tool support will be discussed in the prototype and implementation chapter.

4.6 Case study and evaluation

Now we will use an example to show how to apply CIMOn scheme to reuse and integrate ontology in a context. In the previous chapter, we have presented a modularized ontology knowledge base regarding laundry detergent product of EU Eco-label. It is a very common scenario that a product profile is required to be modeled in the ontology. So, let's suppose that, in some eco-labeling process or evaluation task, knowledge engineers or the domain experts want to use the knowledge in ontology to construct a product profile representation. Detailed parameters of the product example is list in Table 4.1.

Table 4.1: Detailed parameters of heavy-duty laundry detergent product profile example:

property parameter	**value**
product type	liquid
recommended dosage (reference dosage)	20.0 ml/kg wash
sales country	France
weight utility ratio (WUR)	2.0 g/kg wash
critical dilution volume (CDV)	30000.0 l/kg wash
aerobically non-biodegradability (aNBO)	0.5 g/kg wash
anaerobically non-biodegradability (anNBO)	0.5 g/kg wash
known ingredient:	C10-13 linear alkyl benzene sulphate; C8-12 Alkyl ether sulphate; Phosphonate; Sodium Lauroyl Methyl Isethionate; Benzisothiazol; Methylisothiazolinone.

In this product profile building scenario or modeling scenario, we should be aware of that, first, we need the knowledge and information from ontology; second, we possibly need only part of the knowledge of certain ontologies. In the previous chapter, we have provided certain flexibility to ontology knowledge base by introducing modularization and an entity-rule separation. While, sometimes, higher level of flexibility is required, e.g. in the product profile modeling scenario that we have just presented. In this scenario or context, *Module Iso_standards*, *Module Regulation_european_commission* and *Module Commission_decision* can be excluded, because they are relevant as regards to the whole domain but irrelevant for this specific scenario. More over, since there are only six ingredients that are explicitly listed, only a small part of *Module Ghs_hazard_statement*, *Module European_risk_phrases* and *Module Didlist* are needed. Figure 4.9 illustrates how we identify and form a context for this laundry detergent product profile modeling scenario. The first step, relevant ontology modules are identified and chosen. This step requires that knowledge engineers have a basic understanding of the module's content. Second step, the dependencies between modules should be removed, e.g. owl:imports and sub-content or sub-module should be identified. In our example, the whole *Module Laundry_detergent* is needed, thus it is kept entirely. The other three modules are tailored so that only the useful parts are kept. (The star symbol means that the module should be partly used and a sub-module is identified. In this scenario, rule modules are ignored.) At last, the third step, a context involving the necessary content is formed. Complementary relations can be added and edited in the context. In our case, classes with the same name "FunctionalUnit" are defined in both *Module Laundry_detergent* and *Module Didlist*. Axiom *Laundry_detergent:FunctionalUnit owl:equivalentClass Didlist:FunctionalUnit* and other mapping axioms are inserted in the context configuration.

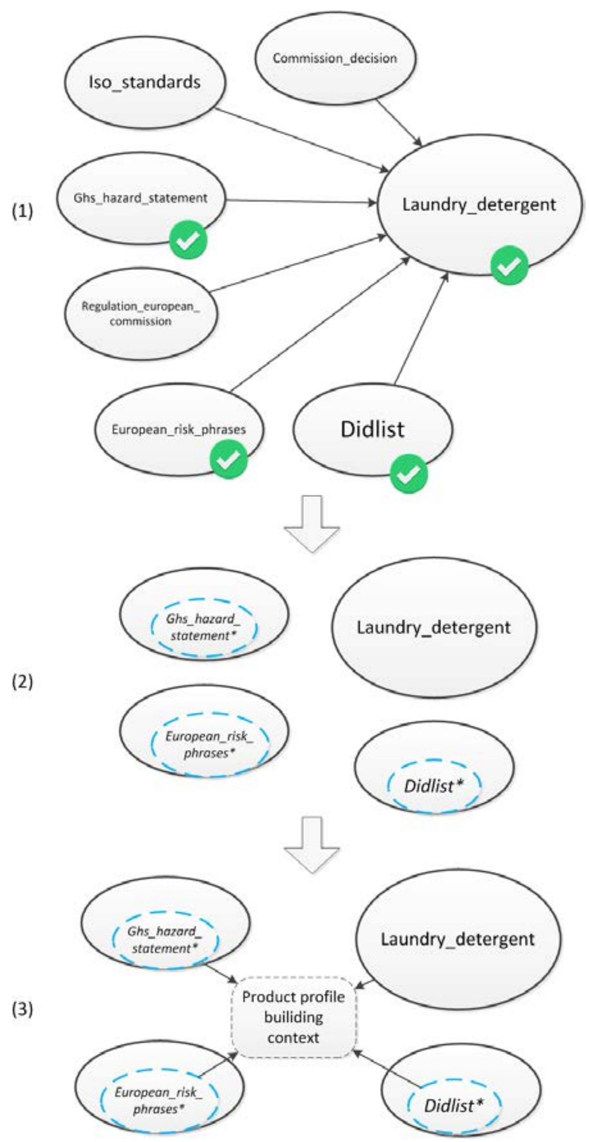


Figure 4.9: A context is identified from modularized ontology knowledge base.

One of the significant benefits of this contextual integration or partly reuse method is that the size of the integration outcome is smaller than the one that uses owl:imports. A better reasoning performance is also obtained with regards to the contextual integration. Table 4.2 is a comparison between traditional owl:imports method between CIMOn method.

Table 4.2: Detailed parameters of heavy-duty laundry detergent product profile example:

	integration by owl:imports	**contextual integration by CIMOn**
Logical axiom count	5049	425
Declaration axioms count	600	198
Class count	68	64
Object property count	52	44
Data property count	21	21
Individual count	459	69
Mean reasoning time (Hermit)	971ms	212ms

In a global point of view, we have also conducted a comparison between our CIMOn solution and other available modular ontology methods. With regards to some criteria, CIMOn is proven to have better performance, while for some other criteria, it does not have obvious advantages. Table 4.3 is the result of our survey and comparison. From the comparison, we can see that CIMOn method has very good ontology dynamics feature thanks to the layered architecture and context configuration. Real-time ontology reuse and integration are possible. Another important advantage of CIMOn method, as its name indicates, is the awareness of context. For each reuse context or integration scenario, independent context configuration is provided. With the help of context, the coupling between modules is greatly weakened as all dependencies are saved in the context rather than original modules. It is also because of these features that CIMOn method has good scalability and re-usability. While, the main weakness of CIMOn could be the very limited heterogeneity and security support. For now, CIMOn only support part of OWL and RDF expressions, the other ontology languages or syntaxes are not supported yet. Whether CIMOn guarantees a conservative extension is still under investigation. There is also a lack of privacy and security consideration. We plan to take into account these issues in our future work. Overall, CIMOn method is a flexible OWL ontology reuse and integration solution.

Table 4.3: An comparison of several popular modular ontology formalisms (T stands for TBox and A for ABox)

	OWL DL	**DDL**	**IDDL**	**P-DL**	**E-Connection**	**CIMOn**
Encapsulation	Partial	Yes	Yes	Yes	Yes	Yes
Re-usability	Fair	Good	Good	Good+	Good-	Good+
Trust and Security	No	No	No	Partial	No	No
Ontology Dynamics	Yes	No	No	No	No	Yes
Loose Coupling	No	Yes	Yes	Yes	Yes-	Yes+
Self-Containment	Partial	Yes	Yes	Partial	Yes	Yes
Scalability	Low	Fair	Fair+	Fair	Low	Fair
Reasoning Support	T and A	T and Partial A	T and A (Under investigation)	T	T	T and A (Under investigation)
Context Awareness	No	Yes	Yes	Yes	Yes	Yes+
Heterogeneity Robustness	Very Limited	Good	Very Good	Limited	Excellent	very Limited
Conservative extension	Partial	Unidentified	Unidentified	Unidentified	No	Under investigation

4.7 CIMOn in ontology engineering practice

In the last section of this chapter, we will discuss how to apply the CIMOn method in realistic ontology engineering practice. We will take several ontology engineering methodologies for example, identify the key tasks for ontology integration or composition, then try to adopt contextual information into them.

Today, even though ontology engineering is relative younger than software engineering, quite many ontology engineering methodologies or methods have been proposed and studies. The interesting thing is that most of them can be traced back to classical software engineering methods. The engineering processes identified in METHONTOLOGY [55] are: specification, knowledge acquisition, conceptualization, integration, implementation, evaluation. It seems alike to the classic water-fall development model which is mostly a sequential development model. If the ontology is not developed from scratch, some ontology modules should be reused. Knowledge engineers should be aware of the ontology reuse from the very beginning of the development, i.e. specification step. Thus, an ontology reuse context or ontology integration context should be identified in the specification step. In this step, besides the traditional competency questions in terms of ontology specification, engineers should ask themselves and answer the question like, what kind of knowledge could I possibly refer to? Is there possibilities that I can reuse some knowledge or ontology? Where can I get access to these knowledge or ontologies? Have these knowledge or ontologies been modularized or not? Which modules do I need? Next, in the knowledge acquisition and conceptualization processes, we can proceed as usual. Then, in the integration and implementation steps, we shall apply CIMOn method and generate the XML implementation for the context. More specifically, at the end of conceptualization, ontology modules should be identified. In addition, we should be aware of which definitions or axioms should be put into context and which ones should be put into modules. In the integration step, the context and the ontology modules will be composed into the ontology artifact. In the final evaluation part, an evaluation of the context in terms of modeling accuracy, reasoning efficiency, etc. is also needed. According to the research described in chapter 2 of [44], METHONTOLOGY is a relative complete method. Instead of linear process, most of methodologies have refinement process where each activity can be repeated several times. In addition to the activities that we have seen in METHONTOLOGY, important activities include maintenance and documentation. While, these two activities are easy to apply a contextual extension.

As stated in [113], modularization and distribution become important trends in terms of ontology engineering in Semantic Web environment. Three approaches can be involved in realizing the modularization of an ontology: ontology partitioning, ontology module extraction and ontology module composition. To our knowledge, our CIMOn method should be cataloged under ontology module composition. However, before putting ontology modules into certain context, these modules should be ready at first. Thus, techniques and activities about ontology partitioning or ontology module extraction should be prior to the application of context.

4.8 Conclusion

In this chapter, at first, we had a review of existent theories about distributed ontology and modular ontology. Various aspects about re-usability, reasoning and modularization are discussed and their problems are identified. Then, we have introduced another important contribution of our work, CIMOn (Contextual Integration for Modular Ontology method), to integrate and reuse different ontology modules. By using "context", extra information needed only when ontologies are to be integrated together can be stored independently. Thus, original ontology modules stay intact and can be reused for different purposes in different contexts at the same time. CIMOn is compatible with owl:imports, and a "filter" is introduced to achieve partly importing which means content of ontology can be partly integrated in context and this feature makes ontology reuse more flexible. Compared to the entity-rule separation pattern that we have introduced in the previous chapter, CIMOn has better usability and flexibility. For now, we have been staying focus on ontology, ontology development and reuse. From the next chapter, we will explore the application and more practical aspects of ontology. In the very next chapter, the decision support process and the corresponding prototype decision support system will be presented.

Chapter 5

A decision support process based on ontology knowledge base

5.1 Introduction

For an eco-label applicant, usually a manufacturer or a service provider, it is easy to provide the required information in whatever formats. However, the difficulties encountered in the evaluating process are representative in decision-making process. To efficiently assess product or service, we need to manipulate different types of voluminous data; take into account different criteria and conduct a multi-criteria analysis; consider different phases of product or service life cycle. Usually, a group of human experts coming from various domains will work together and the evaluating process will take a long time, and errors and conflicts may exist. In addition, the evaluation result is actually a good resource that could have been made better use of.

According to the survey made in [115], among single-standard eco-labels, the most common labels for time required to certification was three to six months, with 37% of respondents falling into this category. The average time to certification across single-standard labels is 4.33 months, the standard deviation is 4.37 months, which indicates that the time to get certificated is quite long, especially for some SMEs (Small and Medium-sized enterprises) and there is still a significant lack of uniformity in the market. Some digitalized management software tools are currently used, however, the evaluation process of eco-labeling certification is still mainly accomplished by the manual inspection or checking of domain experts. If some decision support tool is used in the heavy-duty and knowledge-intensive evaluation process, we believe that the response time will be drastically shortened and the certification cost will be reduced. This is one of the gaps we try to fill in this research. We believe that faster certification with quality and credibility means more opportunities for enterprises and more efficient market. This will also be helpful when eco-labeling is introduced to a fast developing and much bigger market e.g. China and India.

In the previous two chapters, we have presented how to build modularized ontology base. While, the knowledge base itself is not enough for solving problems. We have to find some method to make use of the information stored in the knowledge base. Here in this chapter, in order to better solve problems mentioned above, we propose a decision support process in the scope of eco-label certification i.e. eco-labeling process. The decision support process will take target product's profile as input, compare it with criteria which are stored in the knowledge base composed of ontologies, then generate reasoning result and argumentation or explanation that tells whether the target product can be eco-labeled or not. A significant improvement of our approach compared to traditional decision support systems as mentioned and discussed in [116] and [117] lies in the way the knowledge and data is stored. The knowledge cross-covered in such a decision support process will be represented in modularized ontologies and stored in a knowledge base. Another advantage of our approach is the argumentation or explanation that accompanies the labeling decision. In the light of the argumentation, decision makers can have clearer understanding on how the decision result is made and why. As is illustrated on the left side of Figure 5.1, the objective of our research is to assist and accelerate the evaluation process of eco-labeling to help domain experts make wiser decisions on behalf of the administration and management of eco-labeling. The proposed knowledge base of this system will contribute the reuse of eco-label products knowledge and improve its interoperability with other systems, such as EMS (Environment Management Systems), PLM (Product Lifecycle Management) systems, and ERP (Enterprise Resource Planning) systems. Simultaneously, from the point of view of a producer as an eco-label applicant, such a decision support tool can serve as a simulation tool that will assist the design and validation phases of new product's development, as shown on the right side of Figure 5.1.

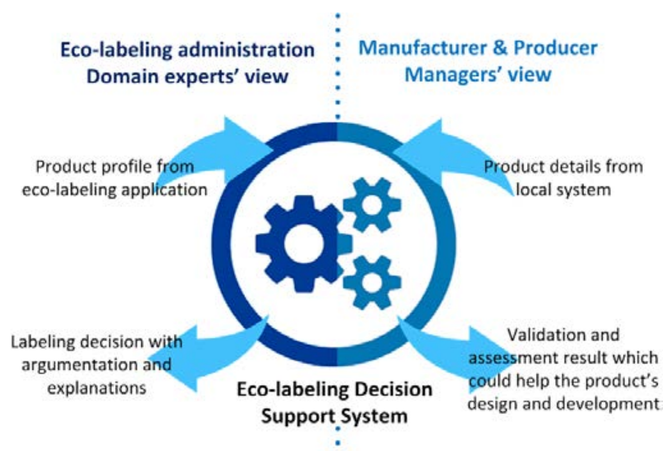


Figure 5.1: Objective and function of the decision support system from both eco-labeling administration and producer's point of view.

EU Eco-label is our study case, since it is still a large and complex labeling system covering dozens of product and service groups and in order to facilitate the development of our decision support system, we will focus only on one product group. Thus, we choose such a laundry detergent product group which is of middle size in the product hierarchy. Another reason is that the eco-labeling for laundry detergent has been carried out for years and we already have successful application cases to be studied and labeled detergent products on shelf to be validated. It should be noted that, the underlying decision support process implemented in the system is not limited to specific product or categories. By generalizing the steps and tasks, replacing the ontologies in the knowledge base, the decision support process can be also applied to other product group.

5.2 State of art

Although some environmental performance assessment methods and tools have been developed, e.g. LCA (Life Cycle Assessment), ERA (Environmental Risk Assessment) and LCC (Life Cycle Cost) mentioned in [118], EDIP 1997 [119], OMNIITOX [120], IMPACT 2002+ [121], etc., most of these methods and tools focus barely on technical analysis. There is a lack of appropriate software implementation support especially in terms of decision support functionality. In the work of [26], a tool to verify the compliance of a product with given norms and standards in terms of the recyclability of industrial products is described. It is shown how the knowledge contained in eco-labeling standards and norms in textual form can be translated into constraints. NIAM/ORM (Object-role Modeling) can be used for formalizing the product data and the verification of the eco-label compliance turns into a CSP (Constraint Satisfaction Problem). In their later work in [27], the authors present how CLAIRE (Combining Logical Assertions, Inheritance, Relations and Entities) language is used to solve the CSP. Formalizing the data and knowledge contained in textual materials into rules is great, but their work serves only to the design phase of the product. The formalism they chose and the CSP translation are too specific to be reused and hard to interoperate with other systems. In [29], development of environmental knowledge management tool capable of providing planners and production managers the knowledge related to the potential environmental impact of the manufacturing choices in a distributed automotive manufacturing scenario is presented. A web-based software solution is developed for customized user query interface, and ontology based knowledge-base is implemented and validated. However, the explanation function or argumentation of the result of aforementioned systems are weak or absent.

Ontology and Semantic Web technologies have been used in DSS (Decision Support System) during the past decade to solve a number of different tasks, such as information integration and sharing, web service annotation and discovery, and knowledge representation and reasoning [31]. However, according to our survey, there is still no software tools that implement eco-labeling decision support by means of inter-connected ontologies, SWRL rules' reasoning and explanation. Our work explores new possibility of ontology application and its engineering practicability. Reasoning based on SWRL rules, especially, is the core mechanism of our decision support process. We expect that our research can be of some help and inspiration to the application of Semantic Web technologies and the assets of our research, especially the ontology knowledge base, could be exploited by other eco-labels and reused in other similar application or systems.

5.3 An overview of the decision support process

Figure 5.2 presents a simplified outline of the eco-labeling decision support process from two different points of view in terms of the eco-labeling administration's domain experts (scenario 1) and the producer & service provider (scenario 2). There are three roles as participants involved in the decision making process: the applicant as manufacturer or service provider who initiates the eco-label application; the domain experts who take the results of the system and make the final decision on whether the product or service is qualified to be eco-labeled or not; and the member country's authorized Competent Body who is responsible for providing guideline and advice to the applicant while the applicant prepares the required product or service's information.

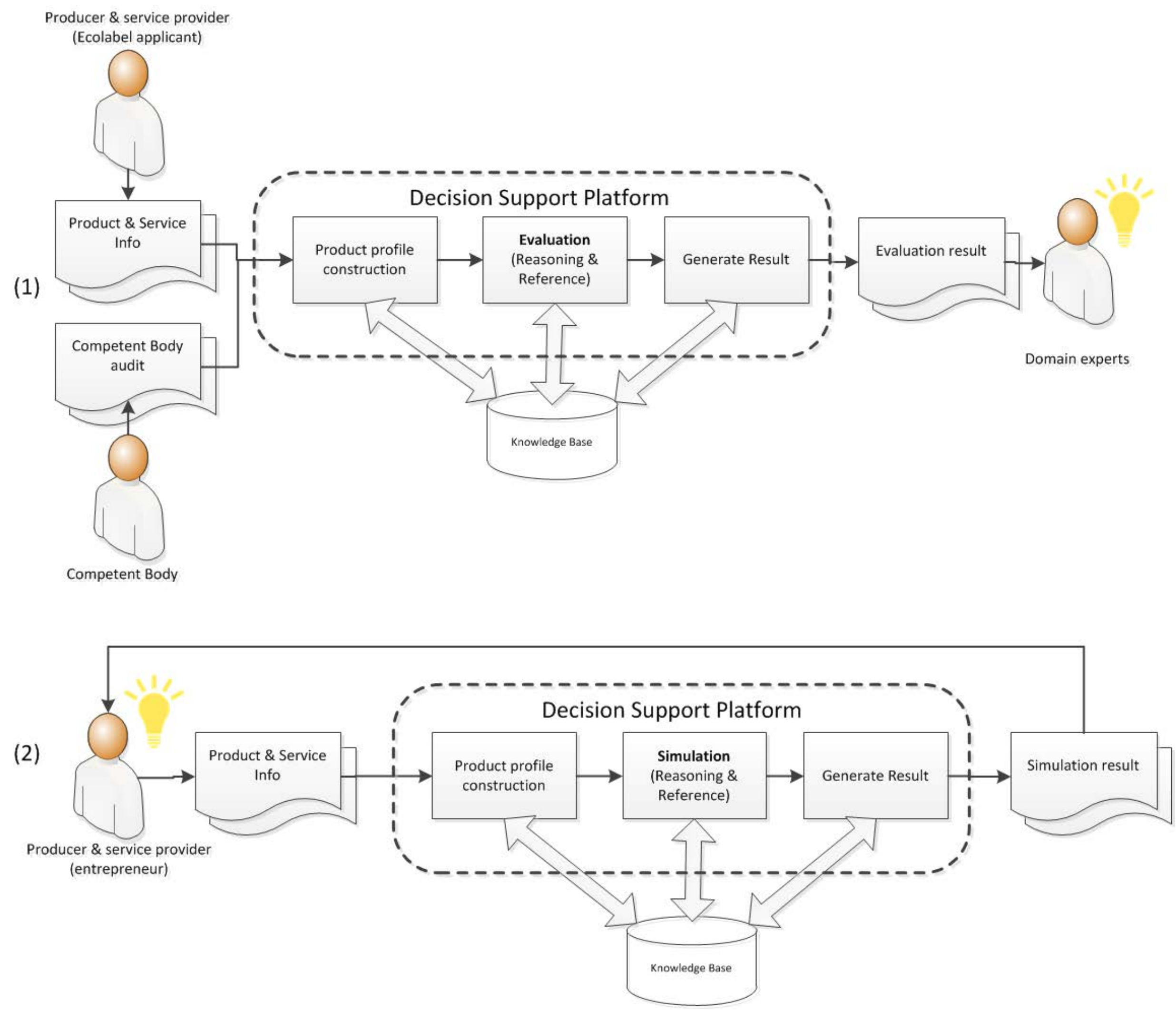


Figure 5.2: Eco-labeling decision support process from two different points of view: (1) domain experts; (2) producer & service provider.

5.4 Architecture and mechanism of the decision support process

Figure 5.3 illustrates detailed mechanism of the eco-labeling decision support process. Meanwhile, it can also be regarded as the architecture or structure of the decision support system based on this decision support process. In a standard evaluation or simulation process, a detailed description of product or service is provided as the key input of the system. The system retrieves concerned information from the product description and builds a machine readable structured document. Then, the structured document will be transferred into a Product Profile Ontology (ABox) in accordance with a Template Ontology (Tbox) retrieved from the ontology knowledge base. Afterwards, the system will select related domain ontologies from the knowledge base. Towards these ontologies, a modularization and refinement formation will proceed in order to gather the very necessary knowledge parts (usually the obligatory criteria rules) to build a merged Criteria Ontology. In the next step, the inference takes place on the combination of Product Profile Ontology and Criteria Ontology, verifying if the Product Profile Ontology that contains description of the product comply with the corresponding criteria. At last, an argumentation generation component will parse and translate the conflicts between Product Profile Ontology and Criteria Ontology so as to generate the final report for human experts review. All the document processing, reasoning or generating process will be supported by a comprehensive knowledge base composed of ontology. The rest part of this section is devoted to describe how we make use of the knowledge base to facilitate a product/service evaluation with traceable argumentation.

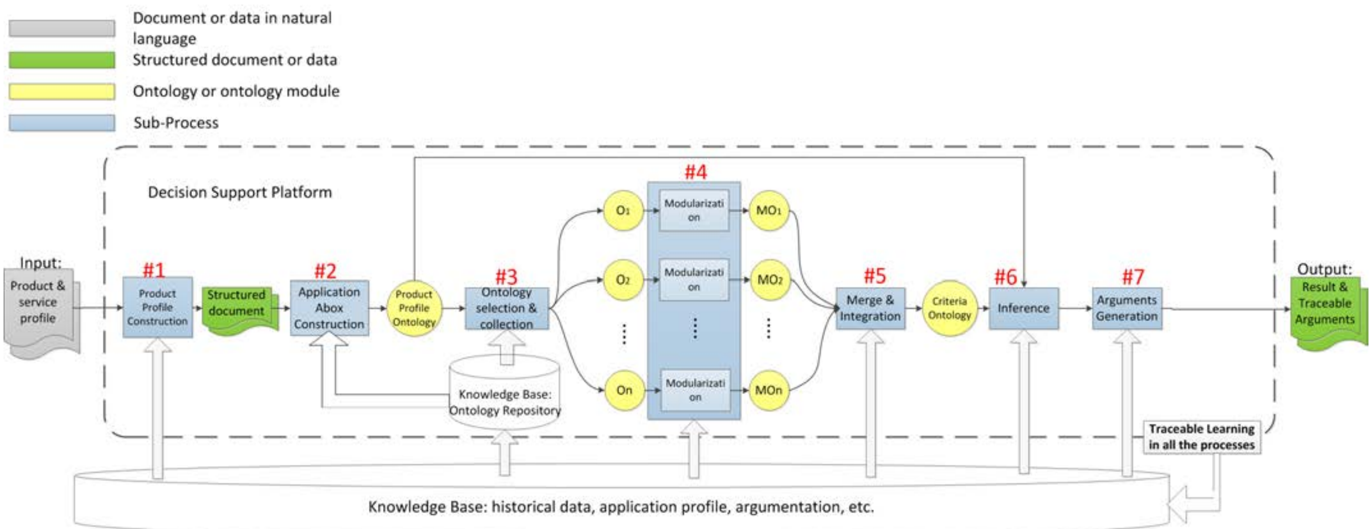


Figure 5.3: Functioning of the decision support platform for eco-labeling

Sub-process #1: Product profile construction This is the very first step of the decision support process and the detailed product or service's profile is required (Figure 5.4). In this step, system will extract useful information from product profile documentation and store them into a structured format. The inputs are the documents provided by the applicant and the output is structured document in XML. If necessary, a NLP module that translates documents to an XML format can be applied here. But the detailed method of translation is not the scope of this work. For now, electronic forms are used to capture product's profile. The information provided by applicant contains not only numeric parameters and data but also product packaging artwork, administrative or legislation declaration, laboratory analysis report. Some are easy to be processed by machine such as all kinds of forms, while some other need human reading and comprehension as what we have discussed in the previous subsection

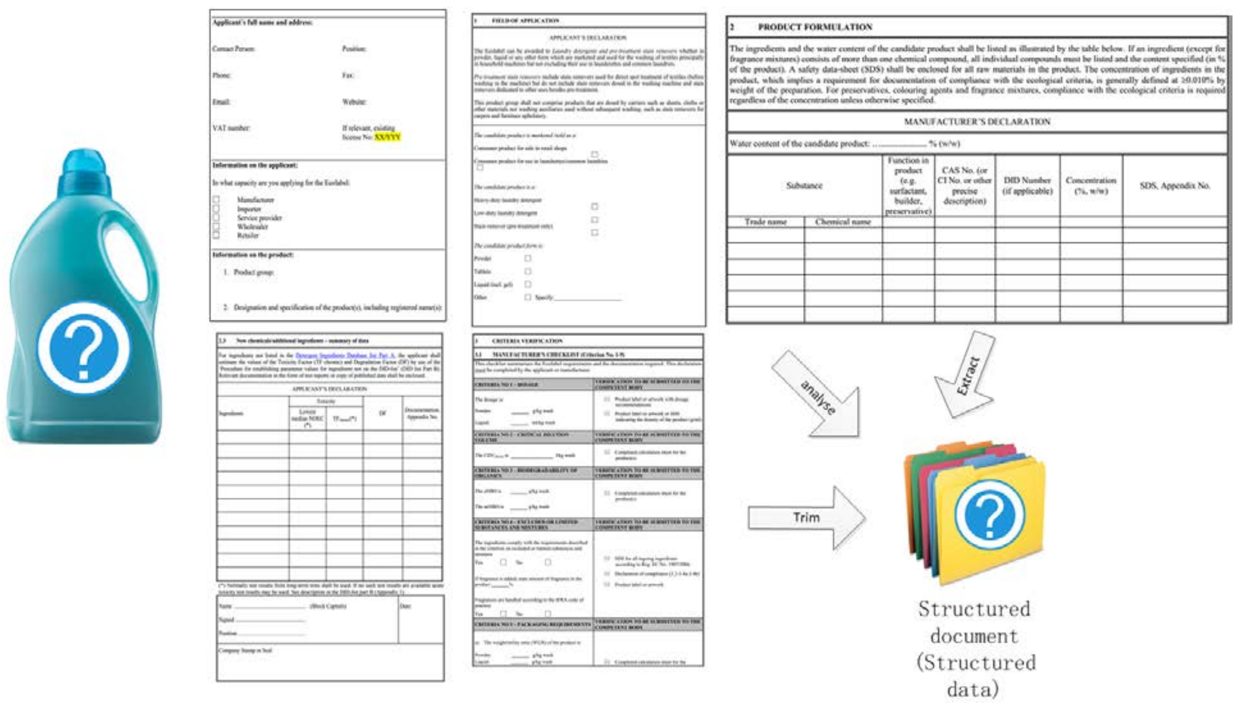


Figure 5.4: Sub-process #1: Product profile construction.

Sub-process #2: Product Profile Ontology Construction This step (Figure 5.5) is very critical to the next steps and sub-processes. As the knowledge base is composed of ontologies of different product groups, we need to choose and transfer or translate the input structured product profile into corresponding ontology assertions. In this translating process, a Template Ontology is required. Template Ontologies are predefined in the ontology repository which is also a part of the knowledge base. A Template Ontology is like a stereotype or meta model of product profile defined in ontology language. It specifies the structure and format of the product's composition, physical & chemical characteristics, and other required information. Usually a Template Ontology is a TBox, in which few instance or individual exist, it mainly specifies the conceptualization of a product or service. The system will send a query on the basis of different product group to the ontology repository, correspondingly, the right template will be retrieved from the repository. Formatted by Template Ontology, the product & service's information will be organized and transferred into an ABox called

Product Profile Ontology.

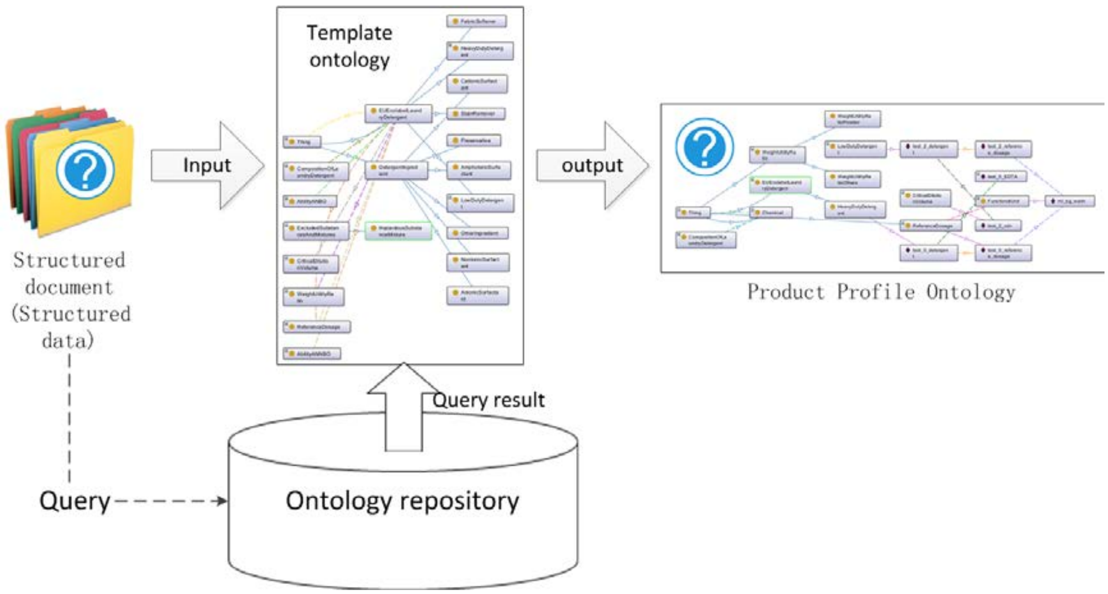


Figure 5.5: Sub-process #2: Product Profile Ontology (ABox) Construction.

Sub-process #3, #4, and #5: Criteria Ontology generation The objective and output of sub-process #3, #4, and #5 (Figure 5.6) is to generate the Criteria Ontology. Based on the semantics of input product & service profile in Product Profile Ontology, ontology or ontology modules will be selected and collected from the knowledge base and the ontology repository. We call this integration of modules as Criteria Ontology, a knowledge component which holds product criteria. Please note the difference between Product Profile Ontology and Criteria Ontology. The former one is mostly a descriptive assertion and instantiation that records the product or service's profile, while a Criteria Ontology is supposed to contain the assessment criteria and guidelines. It is in compliance with the official eco-labeling criteria documents. In a technical point of view, all the SWRL rules should be defined in the Criteria Ontology.

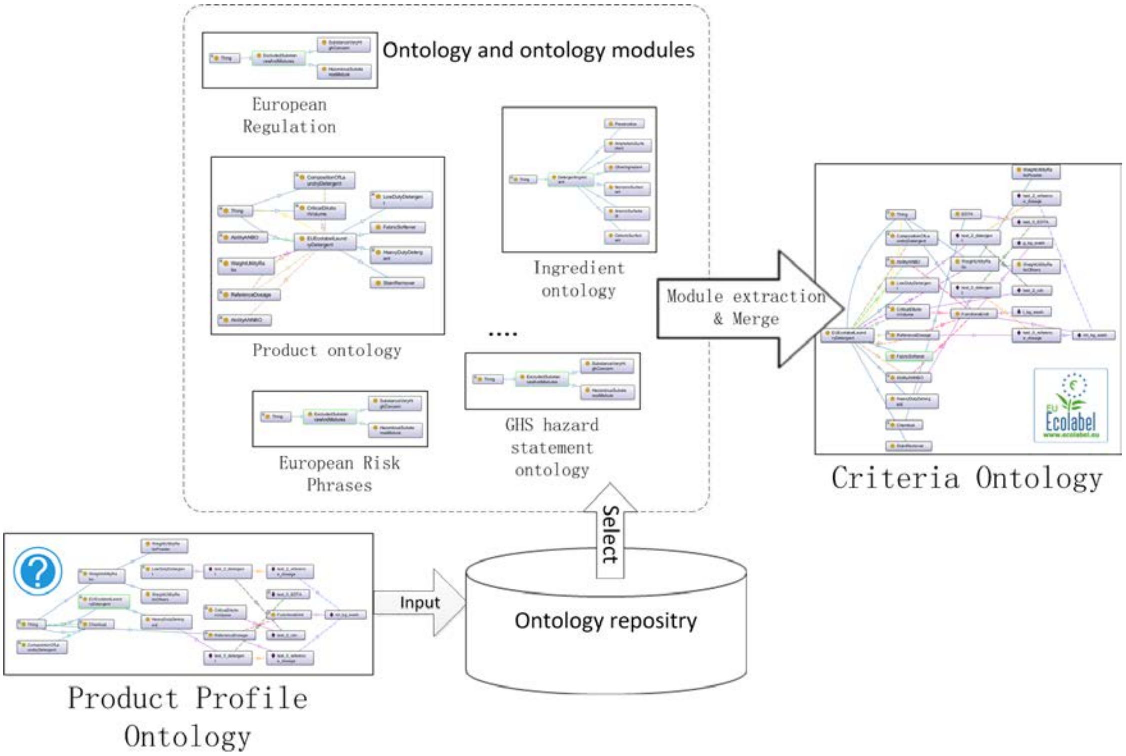


Figure 5.6: Through sub-process #3 to #5 a Criteria Ontology is constructed.

In sub-process #4, modularization may be applied in the ontology. (Instead of generic definition of ontology modularization or modular ontology, "modularization" here means module partition and module extraction.) Because that for single product or service, only part of the ontology repository should be used. Moreover, sometimes even the selected items could be too complex and too big regarding to their structure and volume. If we combine these components together directly, unnecessary knowledge and information will exist in the Criteria Ontology and that will introduce negative influence to the reasoning performance. Thus, an ontology "pruning" is needed. The ideal situation is the system can autonomously select, prune, and merge useful ontology modules.

In sub-process #5, a merge and ontology integration is proceeded to finalize and generate the Criteria Ontology.

Sub-process #6 and #7: Inference & Arguments generation Taking the input Product Profile Ontology and the Criteria Ontology, inference sub-process takes charge to indicate which part in the Product Profile Ontology doesn't comply with the rules defined in the Criteria Ontology. Then, arguments generation component will generate reasons and explanations on why these inconsistency and noncompliance exist. If necessary, a suggestion from the knowledge base trying to solve this noncompliance will be proposed. In addition to the plain inference result in ontology language, the arguments generation component will parse and translate the reasons and explanations into a reading friendly output report for human experts' review. Till here the task of the decision support system is finished and the following procedure will be the experts judge the results and feedback to the applicant.

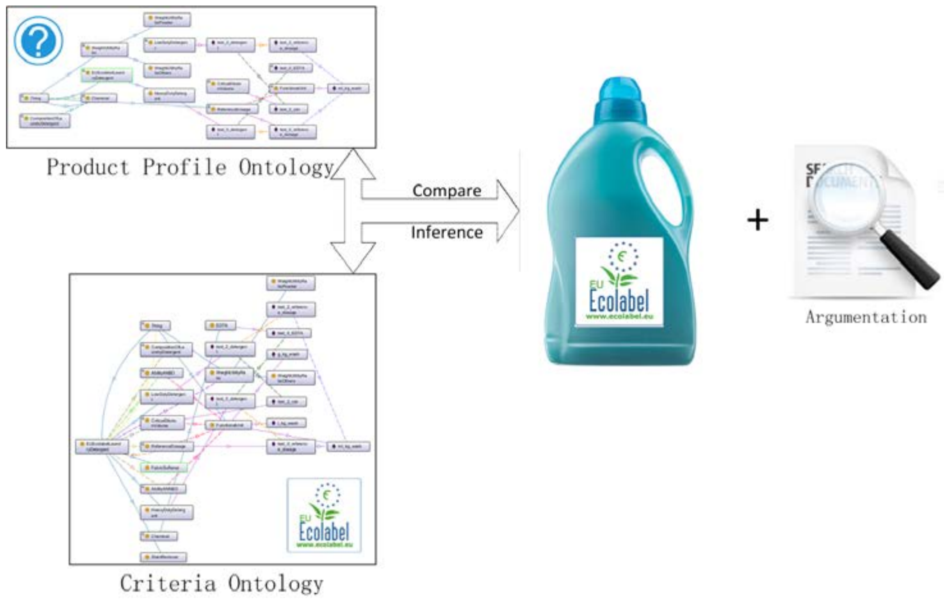


Figure 5.7: Via the comparison and inference applied between the Product Profile Ontology and Criteria Ontology, labeling suggestion and argumentation are obtained.

To finish this section, note that all the sub-processes will be supported by the knowledge base. The knowledge base preserve comprehensive types of data and knowledge. A critical part of the knowledge base is the ontology repository (see Figure5.3), where all the Template Ontologies, Product Profile Ontologies, Criteria Ontologies are kept. The other part is to store all the historical data, application, reasoning result, and argumentation generated during every decision support process. The knowledge base is connected to other data source locally or remotely. To achieve a better interoperability performance, the ontology repository is equipped with public semantic data source accessing interface, which allows the ontology and data stored locally to be accessed by other applications. In the opposite direction, the knowledge base is designed to be able to browse other knowledge base or ontology repository e.g. ChEBI[1], ChEMBL[2], BioPortal[3] and PubChem[4] to acquire extra information. With such an open information sharing mechanism, we guarantee that part of the knowledge base shall be shared. This will be the cornerstone of interoperability when the decision support system is about to cooperate with other systems or is to be integrated into other systems, such as PLM (Production Lifecycle Management) system or EMS (Environment Management System).

[1] https://www.ebi.ac.uk/chebi/init.do
[2] http://www.ebi.ac.uk/rdf/services/chembl/sparql
[3] http://bioportal.bioontology.org/
[4] https://pubchem.ncbi.nlm.nih.gov/search/#collection=compounds

5.5 Case study of EU Eco-labeling on laundry detergent

As we've seen in the state of art section, the EU Eco-label product catalog covers more than thirty product groups which is really a big criteria system. For our primary implementation, we start from single product group to test and validate our decision support process. In our research, laundry detergent product group is chosen to be our study case. Because the criteria volume of laundry detergent is neither too big nor too small. In the market, quite a number of laundry detergent products are already eco-labeled and such a popularity can provide enough successful eco-labeling application cases to be further studied.

A necessary part of the implementation of the decision support system infrastructure is the knowledge base. For our laundry detergent case, a modularized domain ontology is constructed. We have introduced this modularized ontology scheme in chapter 3. This scheme separates "entities" and "rules". The "entities" are the concepts and initial instances referred in the laundry detergent eco-labeling criteria. They are used to describe the conceptualization of the laundry detergent world in the scope of eco-labeling. While, the "rules" are the criterion that reflect people's intention and constraints imposed to the world. This modular design has advantage in terms of reuse and change of the ontology. Since the descriptive part of the ontology does not change very much but the criterion or the "rules" are altered from time to time, once the "entities" and "rules" are separately managed, we can identify one criterion module rapidly and do the updates without disturbing the other modules.

A considerable advantage of using OWL ontology is that the underlying DL (Description Logic) formalism allows reasoning. In this section, we focus on how to apply reasoner to do the reasoning and how to generate the argumentation. We'll see how sub-process 6 and 7 are implemented in the prototype of our decision support system. In Protégé editor, several third-party reasoners exist as plug-ins. In fact, today's reasoners can also stand along as APIs or even independent tools. Since Protégé is an open source project, for almost all its reasoner plug-ins, we can find corresponding APIs that can be integrated into programming language like Java or C++. Actually, the integration of reasoners will be a vital part of the implementation of our decision support system for eco-labeling which is described in the previous section. The originally reasoners carried along by Protégé (the version we used is Protégé 5.0.0 beta 24): FaCT++ is a sound and complete reasoner for $SHOIQ$ (the same description logic underlying OWL-DL) [122]; Pellet is also a sound and complete reasoner which would be very interesting for our research [100]; Hermit [123] works best with our ontology knowledge base as for the SWRL rules, so all the reasoning tasks involved in this work is completed by Hermit (The version we use is 1.3.8.413).

After checking all the criterion in the laundry detergent product, we translated them into SWRL rules. Take the weight/utility ratio part of packaging requirements criterion for example. This part of the criterion, which is quite simple, is to control the weight/utility ratio (WUR) of the product in order to use less packing materials to contain more products. In EU Eco-label criteria document, the details of this criterion is shown in Table 5.1.

Table 5.1: Criterion of packaging requirements. The weight/utility (WUR) of the product shall not exceed the following values.

Product type	**WUR**
Powders	1.2 g/kg wash
Others (e.g. liquids, gels, tablets, capsules)	1.5 g/kg wash

Based on our criteria ontology, this criterion can be translated to SWRL rules like this:

1. $CandidateLaundryDetergent(?a), hasDetergentProductType(?a, powder),$ $hasWeightUtilityRatio(?a, ?wur), hasValue(?wur, ?wur\_value), xsd : double[> “1.2''\hat{}\hat{}xsd : double](?wur\_value) \rightarrow RejectedDetergent(?a)$

2. $CandidateLaundryDetergent(?a), hasDetergentProductType(?a, liquid),$ $hasWeightUtilityRatio(?a, ?wur), hasValue(?wur, ?wur\_value), xsd : double[> “1.2''\hat{}\hat{}xsd : double](?wur\_value) \rightarrow RejectedDetergent(?a)$

3. $CandidateLaundryDetergent(?a), hasDetergentProductType(?a, gel),$ $hasWeightUtilityRatio(?a, ?wur), hasValue(?wur, ?wur\_value), xsd : double[> “1.2''\hat{}\hat{}xsd : double](?wur\_value) \rightarrow RejectedDetergent(?a)$

4. $CandidateLaundryDetergent(?a), hasDetergentProductType(?a, tablet),$ $hasWeightUtilityRatio(?a, ?wur), hasValue(?wur, ?wur\_value), xsd : double[> “1.2''\hat{}\hat{}xsd : double](?wur\_value) \rightarrow RejectedDetergent(?a)$

5. $CandidateLaundryDetergent(?a), hasDetergentProductType(?a, capsule),$ $hasWeightUtilityRatio(?a, ?wur), hasValue(?wur, ?wur\_value), xsd : double[> “1.2''\hat{}\hat{}xsd : double](?wur\_value) \rightarrow RejectedDetergent(?a)$

The basic idea for almost all the criteria rules is by introducing concepts called “Rejected-Detergent” and “CandidateLaundryDetergent” in the ontology. At the start of the decision support process, we input product profile as instance of the “CandidateLaundryDetergent”, and at this moment, we still don’t know if this product respects the criteria or not. Once the reasoning process is started, these SWRL rules are applied upon the Product Profile Ontology. As long as the profile of some detergent product does not comply with the criteria rules, this product should be classified as an instance of “RejectedDetergent”. In other words, this class is treated as the “objective” of the reasoning task. After the reasoning, if a product instance is classified under the type of “RejectedDetergent”, we assert that this product doesn’t comply with EU Eco-label criteria standard and the eco-label should not be approved.

As we interpret these rules, we note that the concepts (classes) and relation (properties) appear in the rules are already defined in the Criteria Ontology. They also appear in the Product Profile Ontology and the Template Ontology. Actually, the class and relation hierarchy of Product Profile Ontology and Criteria Ontology must be consistent to allow the rules in Criteria Ontology to be applied. *CandidateLaundryDetergent* is a class that generalizes all the detergent product before the decision making process. *hasDetergentProductType* is an object property specifying the product type, its domain is *CandidateLaundryDetergent*, the range is class *DetergentProductType*. Under the type of *DetergentProductType*, 5 different product type instances (*powder, liquid, gel, tablet, capsule*) are defined in both Product Profile Ontology and Criteria Ontology to represent different product format. *hasWeightUtilityRatio* is an object property specifying a product's weight/utility ratio parameter, its domain is *CandidateLaundryDetergent*, its range is *WeightUtilityRatio*. *hasValue* is a general data property specifying quantitative parameter of certain class. In SWRL rules, we put a question mark before a name to indicate that is a variable.

All the reasoning result and new inference of this Product Profile Ontology will be stored in the knowledge base as reference cases for further reuse or review. Thus, the knowledge base for the decision support system will eventually be composed mainly by two parts: an ontology repository that stores all kinds of EU Eco-labeling products' ontologies in modularized way; and a historical cases repository that reserves all their reasoning results and arguments.

The rest part of this section, a simple product profile example will be given to show how explanation and argumentation are generated at the end of reasoning. In our prototype system, the reasoning is implemented by OWL API [5] and the reasoner is Hermit [6]. In this example, we assume that it is a low-duty laundry detergent which is produced by Procter & Gamble and marketed in France. Actually, one product could be marketed across several European countries at the same time. Table 5.2 shows the product's parameters in details.

Table 5.2: Detailed parameters of product profile example: *low-duty laundry detergent example No.0*

property parameter	**value**	**criteria value**
product type	liquid	
sales country	France	
recommended dosage (reference dosage)	15.0 ml/kg wash	17.0 ml/kg wash
weight utility ratio (WUR)	1.4 g/kg wash	1.5 g/kg wash
critical dilution volume (CDV)	23000.0 l/kg wash	20000.0 l/kg wash
aerobically non-biodegradability (aNBO)	0.25 g/kg wash	0.30 g/kg wash
anaerobically non-biodegradability (anNBO)	0.28 g/kg wash	0.30 g/kg wash
known ingredient	Acetic acid; C8-18-Amphoacetates; EDTA	

According to the criteria, the only two known ingredients do not have any hazard code, neither are they in the list of excluded or limited substances, which means they are good to be added into laundry detergent products. However, this product contains one kind of the forbidden substance *EDTA*, also the CDV value exceeds the criteria value, so it should be considered as a "RejectedDetergent". Figure 5.8 is the result from our decision support system prototype program. From the result showing in the console, multiple explanations are presented. The text in blue records all the 6 explanations the reasoner generates. The

[5] http://owlapi.sourceforge.net/
[6] http://www.hermit-reasoner.com/

Hermit reasoner's explanation is originally written in Manchester Syntax[7], which may not be easy to be understood by common user. We've tried to replace some of these Manchester Syntax into plain English. For example, we replaced term *type* in Manchester Syntax with a short phrase *is a type of* in order to increase the explanation's readability. As we can see in Figure 5.8, the first explanation indicates that this product's parameters violate the rule:

$$hasCriticalDilutionVolume(?detergent, ?cdv), double[> \text{“}20000.0\text{''}\char`\^\char`\^double](?cdv\_value), hasValue(?cdv, ?cdv\_value), LowDutyDetergent(?detergent) \rightarrow RejectedDetergent(?detergent)$$

This rule specifies the Critical Dilution Volume of a low duty laundry detergent product. We can interpret this rule like this, if variable *detergent* is a *LowDutyDetergent*, *detergent* has Critical Dilution Volume parameter *cdv*, variable *cdv*'s value is *cdv_value* which is more than 20000.0, then variable *detergent* can be classified to be a *RejectedDetergent*. In other words, this product will not be certificated to be eco-labeled.

Except the first explanation in Figure 5.8, the other explanations concern the rule:

$$hasIngredient(?detergent, ?ingredient), EDTA(?ingredient), CandidateLaundryDetergent(?detergent) \rightarrow RejectedDetergent(?detergent)$$

This rule specifies one of the forbidden ingredients that should not be added into the eco-labeled detergent product. In human language, this rule prescribes that: if *detergent* has ingredient as variable *ingredient*, and *ingredient* is a kind of *EDTA*, then this *detergent* is rejected.

[7]https://www.w3.org/TR/owl2-manchester-syntax/

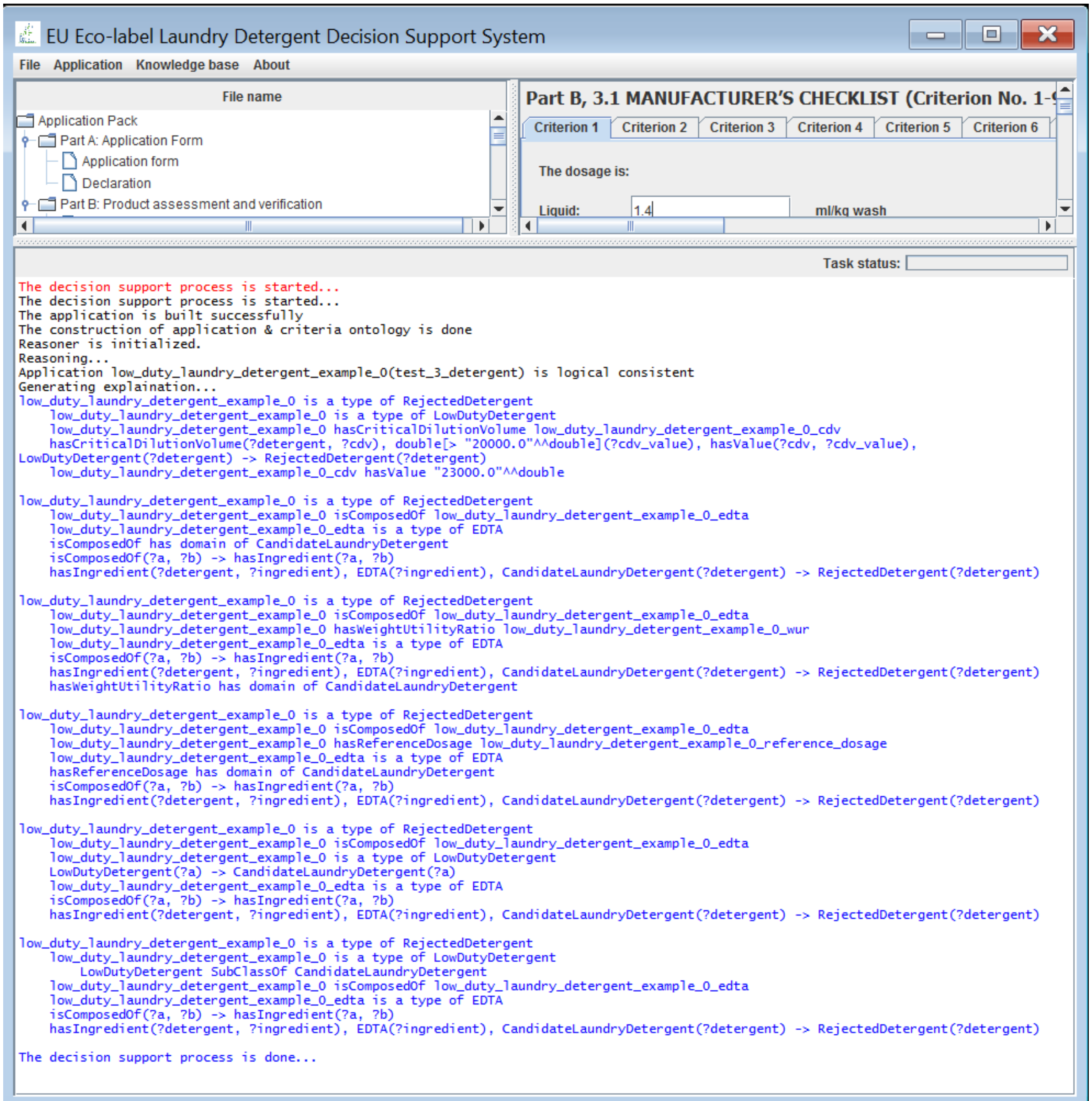


Figure 5.8: Reasoning result of the low-duty laundry detergent example.

About the implementation of the decision support system prototype and other technical details, we will discuss more in the next chapter.

5.6 Conclusion

In order to democratize eco-labeled products and services to achieve a more sustainable economics, a better eco-labeling process is needed. In this chapter, we proposed a decision support process trying to improve and accelerate the evaluation for eco-labeling to help various stakeholders to make wiser decisions as well as to share knowledge and experience. A low-duty laundry detergent product study case was presented to validate that the decision support process generates explanation and argumentation in accordance to the EU Eco-label criteria. The underlying decision support process implemented in the system is not limited to specific product or categories. By generalizing the steps and tasks, replacing the ontologies in the knowledge base, the decision support process and the system can be applied to any other product group. In the next chapter, we will present more technical details about the implementation of the decision support system.

Chapter 6

Prototype and implementation

6.1 Introduction

In previous chapters, a modular ontology knowledge base, the decision support process and CIMOn method are represented respectively. This chapter is more about the implementation of our theory and proposition. More practical and engineering aspects will be discussed. In the first section, we will demonstrate how to realize the CIMOn method with the support of a plug-in developed in Protégé which is a popular ontology editor. The second section is about a standalone decision support system prototype based on the knowledge base and decision support process that we have described in chapter 3, 4 and 5. Both of the two implementation are developed by Java and other open source APIs. Via implementation and result analysis, we aim to validate our theory.

6.2 Context editor plug-in for Protege

6.2.1 A survey about ontology editor and plug-ins

Up to now, to our knowledge, there are already quite several ontology editors or development tools available. Although they are still not as powerful and mature as IDE for software development, they provide various engineering support for knowledge acquisition and ontology development. The most popular ontology editor is Protégé[1] which was initially developed by Stanford University. As an open source project, today's Protégé is more like a platform other than a stand-alone tool. Extensions and plug-ins are encouraged to be developed and integrated freely. NeOn toolkit[2] is also a powerful ontology editor which is quite similar

[1] http://protege.stanford.edu/
[2] http://neon-toolkit.org/wiki/Main_Page.html

to Protégé. Quite many plug-ins are also available in the toolkit. A complete engineering methodology which covers various aspects about ontology life-cycle was also proposed along with the tool [113]. However, it seems that NeOn toolkit is becoming less popular than Protégé in recent days. NeOn toolkit along with many its plug-ins have no update since around 2011 (When this thesis is being written, the last update and release of NeOn toolkit is 12 December 2011). Topbraid composer[3] is a visual modeling environment from industry experts for creating and managing domain models and ontologies in the Semantic Web standards RDF, RDFS and OWL. It is a commercial product developed by TopQuadrant, Inc. According to the specification of this tool, it supports not only ontology manipulation, but also a more powerful linked data and SPARQL query support. Interestingly, it provides as well support of interoperability with UML, XML Schema and databases. There is a list of available ontology editors that can be found on W3C wiki[4]. Another longer list is about ontology related tools.[5] According to our survey and research, most of these tools are research projects or prototypes. Very few have been commercialized like Topbraid composer. The most popular platform or tool is Protégé. Quite amount of these editors or tools are not maintained any more. While, it is still happy to see so much efforts have been done in ontology engineering. We believe that those promising ones will grow and better editors, tools even IDEs are expected.

As we have introduced in the first paragraph, the most popular two ontology tools or platforms are Protégé and NeOn. As far as we can see, Protégé seems more active than NeOn in recent days. There is a Protégé Plugin Library on the website of Protégé wiki [6], and there are 114 plug-ins registered when this thesis is being written. Abundant functions can be found in these plug-ins, while none of them provide the same or similar contextual integration or composition as what we have in CIMOn. Our research work and development aim to contribute to the Protégé plug-in community as well as the domain of ontology engineering.

Based on the survey presented above, we have conducted a requirement analysis and a requirement specification is generated. It is decided that a visual Protégé plug-in should be developed. This plug-in or this tool should be an extension to the original Protégé release. It should not be an extension to some already developed artifact. The most basic function it should provide is the editing of context configuration file. It should allow user select the ontologies he or she needs, specify the root of the context dependency graph, and apply *binaryMappingBundle* between any pair of ontologies. As for each *binaryMappingBundle*, the tool should support fine granularity manipulation e.g. adding and deleting relation between entities. By using this tool, users can save or open context configuration XML files from local system. As we have stated in Chapter 4, CIMOn method integrates multiple ontologies by using extra information stored in the context. At last, the contextual integration of composition is also like a virtual ontology which is made up with original ontologies as well as the context. Our tool will provide some advanced functions such as generation and

[3]http://www.topquadrant.com/tools/IDE-topbraid-composer-maestro-edition/
[4]https://www.w3.org/wiki/Ontology_editors
[5]http://wiki.opensemanticframework.org/index.php/Ontology_Tools
[6]https://protegewiki.stanford.edu/wiki/Protege_Plugin_Library

materialization of the contextual integration. Moreover, before the generation of virtual ontology integration, basic reasoning check will be provided too.

An important reason why Protégé has been chosen as our target platform is because Protégé is already a very powerful ontology editing tool. It support several ontology syntaxes and formats. A big advantage of the usability of our tool depends on the fact that the user can browse the integrated ontologies simultaneously when he is editing the context. Actually, the awareness of details in terms of integrated ontologies will contribute greatly to a context of good quality.

This tool will support relations in OWL/XML syntax. These relations are basic OWL 2 modeling expression elements such as "SubClassOf", "ClassAssertion", "EquivalentClasses", "DisjointClasses", "ObjectPropertyAssertion", "NegativeObjectPropertyAssertion", "SubObjectPropertyOf", "ObjectPropertyDomain", "ObjectPropertyRange", "DifferentIndividuals", "SameIndividual", "DataPropertyAssertion", "NegativeDataPropertyAssertion", "DataPropertyDomain", "DataPropertyRange". Besides these original OWL semantics, some third party relations are also supported. These relations are selected form OBO Relations Ontology [7], namely "aligned with", "depends on", "differs in", "has function", "contains", "derives from", "develops from", "has host", "visits", "input of", "output of", "interacts with", "located in", "has part". These relations are simple relations without domain or range specification. The only semantics that they have depend on the linguistic meaning of their label. Our tool will adopt an extensible design that allows other more third party relations to be included.

This context editor tool will be developed in Java and be made in a plug-in running on Protégé 5.0 and plus for Windows platform. A Jar distribution will be realized. This tool will be visualized as a tab in the window of Protégé. Any function of the newly developed plug-in should not affect the original functions of Protégé.

6.2.2 Plug-in design

Since we have decided to implement the tool as Protégé plug-in, a basic understanding of Protégé mechanism is very necessary. Fortunately, Protégé is an open source project, useful specification and instruction can be found on the official website Protégé wiki[8] as well as GitHub[9]. Especially the latter one, we have checked out the source code of an example plug-in along with the whole source code of Protégé. They are extremely helpful and greatly boost our development.

Before starting coding, a source code inspection is done. [10] While, this inspection is not

[7] http://obofoundry.org/ontology/ro.html

[8] https://protegewiki.stanford.edu/wiki/PluginAnatomy

[9] https://github.com/protegeproject/protege-plugin-examples

[10] We want to show our great respect and appreciate to all the developers and everyone that contributed to Protégé's development and maintenance, especially Mr.Matthew Horridge who did a lot of original development and left inspiring comments in the source code. He also answered and fixed some bug reports in the

complete as today's Protégé is already quite big and complex, we could not afford a code reading that covers the whole project in our limited research schedule. In spite of that, some critical classes and functions are inspected, especially those parts about plug-in. Now, we will present the basic mechanism of Protégé by explaining a basic GUI of class tab. Though similar explanation or tutorial may be found in open source community or other forums, we would like to introduce this in our context. We hope this introduction would be useful to other researchers and developers.

As illustrated in Figure 6.1, which is a basic GUI of Protégé after an ontology is loaded and class tab is chosen. Like most of today's desktop application, a menu bar with commend items is provided, e.g. "File", "Edit", "View", etc. Below the menu bar, there is an ontology chooser which is a combo box menu. This combo box ontology chooser is very important for Protégé. It allows Protégé to open several ontologies at the same time and switch between them. The example ontology opened in the figure is *ro.owl* (RO [11] is a collection of relations intended primarily for standardization across ontologies in the OBO Foundry [12] and wider OBO library. It incorporates ROCore upper-level relations such as part of as well as biology-specific relationship types such as develops from.) Again, below the ontology chooser lay various tabs. In the chosen tab of "Classes", 3 divisions are displayed and actually 5 sub-tabs exist. (According to the specification of Protégé's source code, these sub-tabs are also called views.) On the left side, Class hierarchy view and Class hierarchy view (inferred) are overlapped; the right upper side, Annotation view and Usage view are overlapped; at last, on the right bottom side, Description view is located. When a class item is clicked in the Class hierarchy view on the left side, relevant content will be displayed in the views on the right side. Most of the other upper level tabs follow the same pattern.

community.

[11] http://obofoundry.org/ontology/ro.html

[12] http://obofoundry.org/

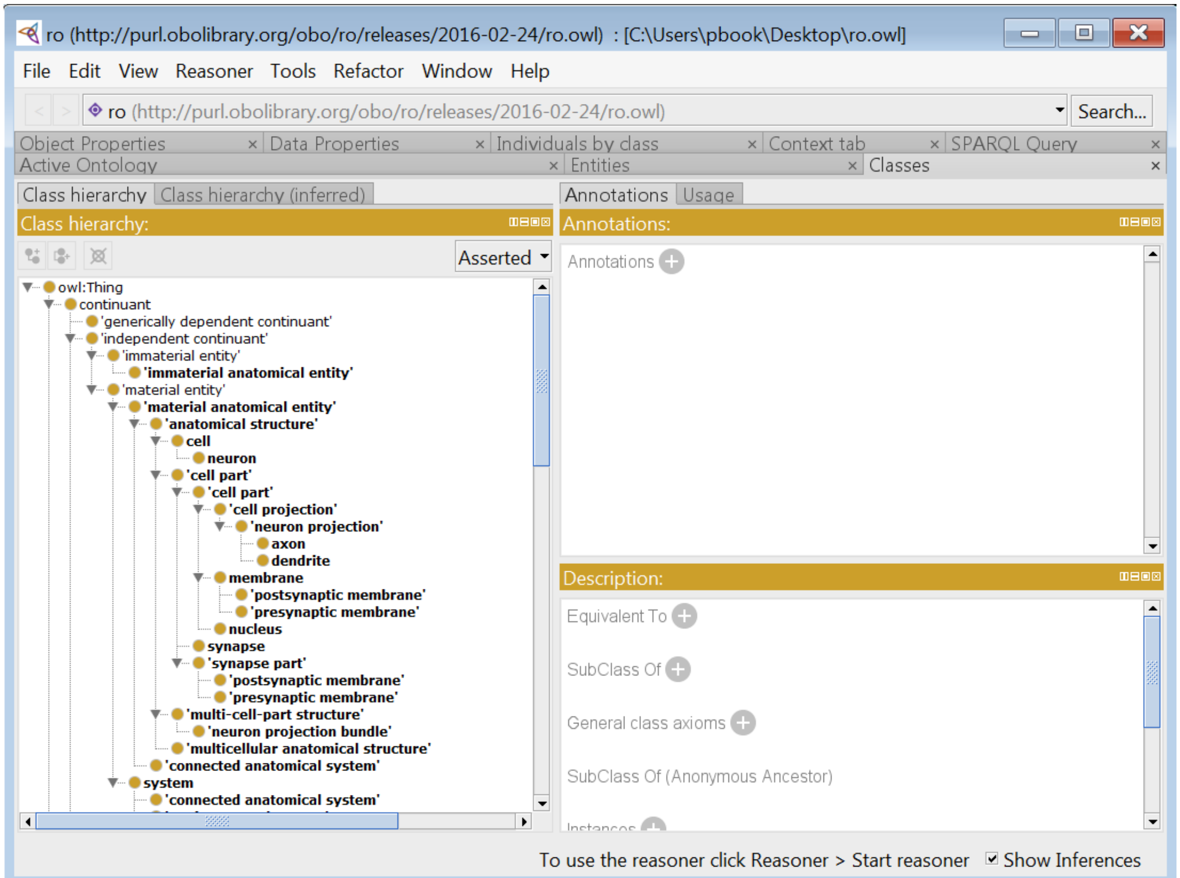


Figure 6.1: Protégé class tab

It is very interesting to see that, this GUI is also implemented in a modularity principle. If we check the source code of the Protégé, a *plugin.xml* file can be found under *$PROTEGE_SOURCE_ROOT$/protege-editor-owl/src/main/resources/*. This file is about the configuration of both the visible components (tabs, views, menus, entity renderers, etc.) and invisible components (actions, etc.) of Protégé GUI. Figure 6.2 is a fragment of this configuration file. This configuration fragment indicates basic parameters for the Classes Tab. We learn from this configuration that the class that implements this tab is *org.protege.editor.owl.ui.OWLWorkspaceViewsTab*, while the visible components making up this tab is configured in file *viewconfig-classestab.xml*. Figure 6.3 is the content of *viewconfig-classestab.xml*. It is in this view configuration file that the layout of tab is specified. We can see from this view configuration file that the 5 view components (Class hierarchy with id *OWLAssertedClassHierarchy*, Class hierarchy (inferred) with id *InferredOWLClassHierarchy*, Annotations with id *OWLClassAnnotations*, Usage with id *OWLClassUsageView* and Description with id *OWLClassDescription*) that show in Figure 6.1 are recorded. Back to *plugin.xml*, all the corresponding classes that are related to the view components of Classes tab (Figure 6.1) will be retrieved, e.g. the class that implements OWLAssertedClassHierarchy is *org.protege.editor.owl.ui.view.cls.ToldOWLClassHierarchyViewComponent*. With the help of Eclipse, all classes that implement these components will be retrieved and a class hierarchy can be constructed.

```xml
<!-- Classes Tab -->

<extension id="OWLClassesTab"
           point="org.protege.editor.core.application.WorkspaceTab">
    <label value="Classes"/>
    <class value="org.protege.editor.owl.ui.OWLWorkspaceViewsTab"/>
    <editorKitId value="OWLEditorKit"/>
    <index value="C"/>
    <defaultViewConfigFileName value="viewconfig-classestab.xml"/>
</extension>
```

Figure 6.2: Protégé Classes tab plugin config fragment

```xml
  <?xml version="1.0" encoding="UTF-8"?>
- <layout>
  - <VSNode splits="0.3 0.7">
    - <CNode>
      - <Component label="Class hierarchy">
          <Property value="org.protege.editor.owl.OWLAssertedClassHierarchy" id="pluginId"/>
        </Component>
      - <Component label="Class hierarchy (inferred)">
          <Property value="org.protege.editor.owl.InferredOWLClassHierarchy" id="pluginId"/>
        </Component>
      </CNode>
    - <HSNode splits="0.375 0.625">
      - <CNode>
        - <Component label="Annotations">
            <Property value="org.protege.editor.owl.OWLClassAnnotations" id="pluginId"/>
          </Component>
        - <Component label="Usage">
            <Property value="org.protege.editor.owl.OWLClassUsageView" id="pluginId"/>
          </Component>
        </CNode>
      - <CNode>
        - <Component label="Description">
            <Property value="org.protege.editor.owl.OWLClassDescription" id="pluginId"/>
          </Component>
        </CNode>
      </HSNode>
    </VSNode>
  </layout>
```

Figure 6.3: viewconfig-classestab.xml

Figure 6.4 is the top level class diagram of the main classes that implement Classes tab. Attributes and functions are not included because of the space limit. Only class names are illustrated. The five view components implementing classes are listed in the very bottom of this class diagram. On the top of this figure, we can see that all the visible components classes are generalized from *JComponent* class which is part of Java Swing. As the functions of Protégé increase, more and more concrete abstraction are generated until the component implementing classes are reached. To our knowledge, *ViewComponent* is a very important class because many basic functions of any view component are defined in it. *Workspace* class is also very important because it contains all important information about Protégé, and unique instance will be initialized every time Protégé is launched. *WorkspaceTab* is the basic abstraction of tab component. Please note that, the classes shown in this figure seem to be only small part of Protégé. This figure only show the branch of Classes tab. There are about a dozen of tabs available. We can not show them all in single class diagram. Here we only take Classes tab for an example to explain the pattern and structure of Protégé's class hierarchy.

As we have claimed in the requirement specification section, our tool shall be an extension of Protégé editor, any change will do harm to the scalability of Protégé. Therefore, an extension design is chosen and Figure 6.5 is the scheme of the design for our plug-in. Class *ContextSkeletonView* is generated from *AbstractOWLViewComponent*. This inherit allows us to customize our own view component. *ContextSkeletonView* is the main frame of our plug-in GUI and three key sub-components are *ContextEditorView*, *SourceOntologyView*, and *TargetOntologyView*. Recall that in Chapter 4, the relations take place between any ontology modules will be wrapped into a *BinaryMappingBundle*. So here our design is that *SourceOntologyView* will be used to browse the ontology module that its bundle starts from. The other one, *TargetOntologyView*, is responsible to browse the module where this bundle ends. *ContextEditorView* will be the editing area for every relation or mapping. All these three views have reference to the container i.e. *ContextSkeletonView*. The *ContextSkeleton-VieContextSkeletonVieww* container will be initialized in *ContextTab* via new configuration added in *plugin.xml*. *OntologyView* is a generalization of JPanel, in other words, *OntologyView* extends JPanel which is part of Java Swing.

This extensive design guarantees that no change will be applied in the original Protégé code which follows the open/closed principle [124]. In the next section, we will present how to implement this design in details and demonstrate the result.

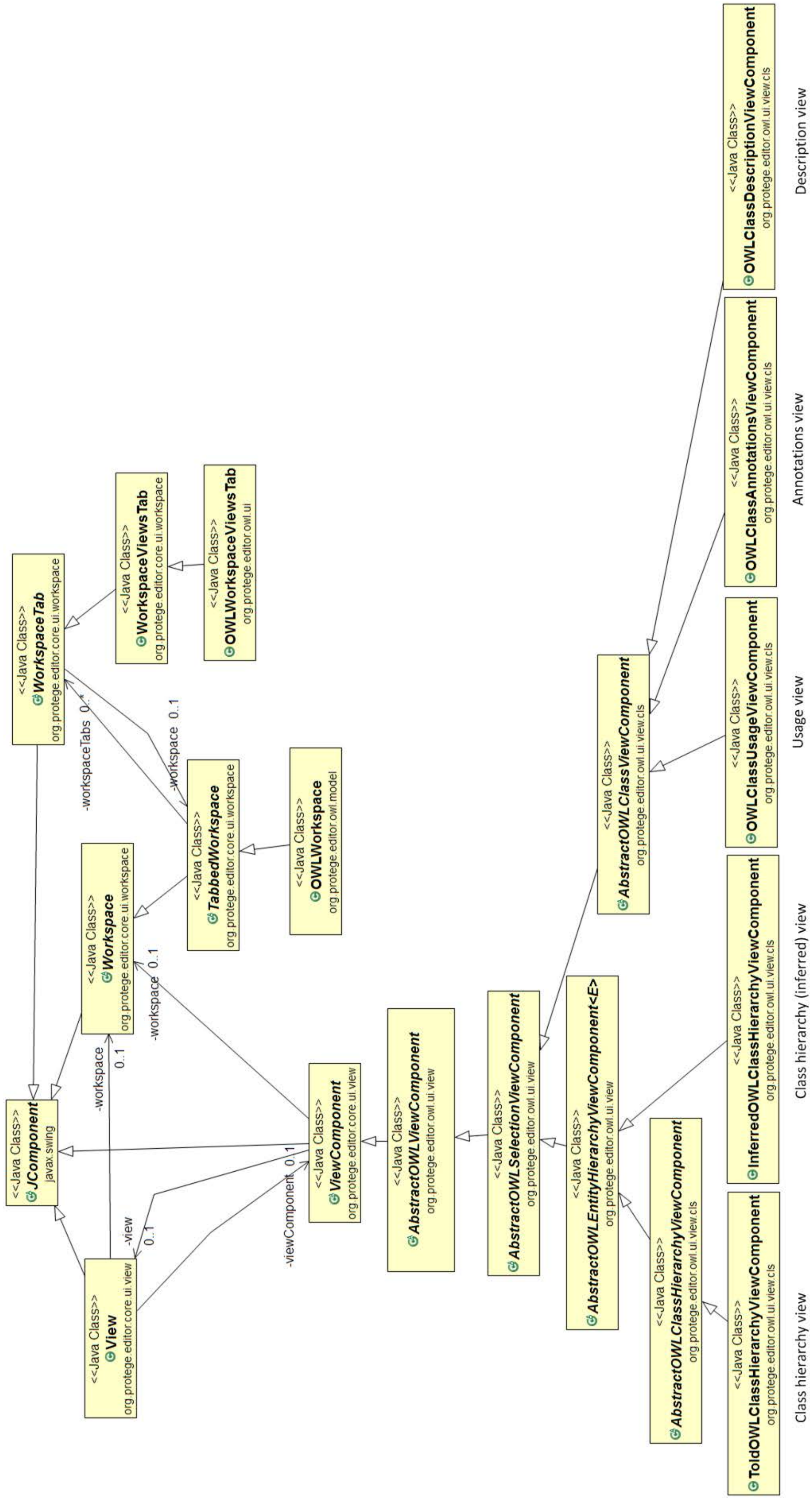


Figure 6.4: Main classes illustrating the architecture of Protégé (Taking Classes tab as example).

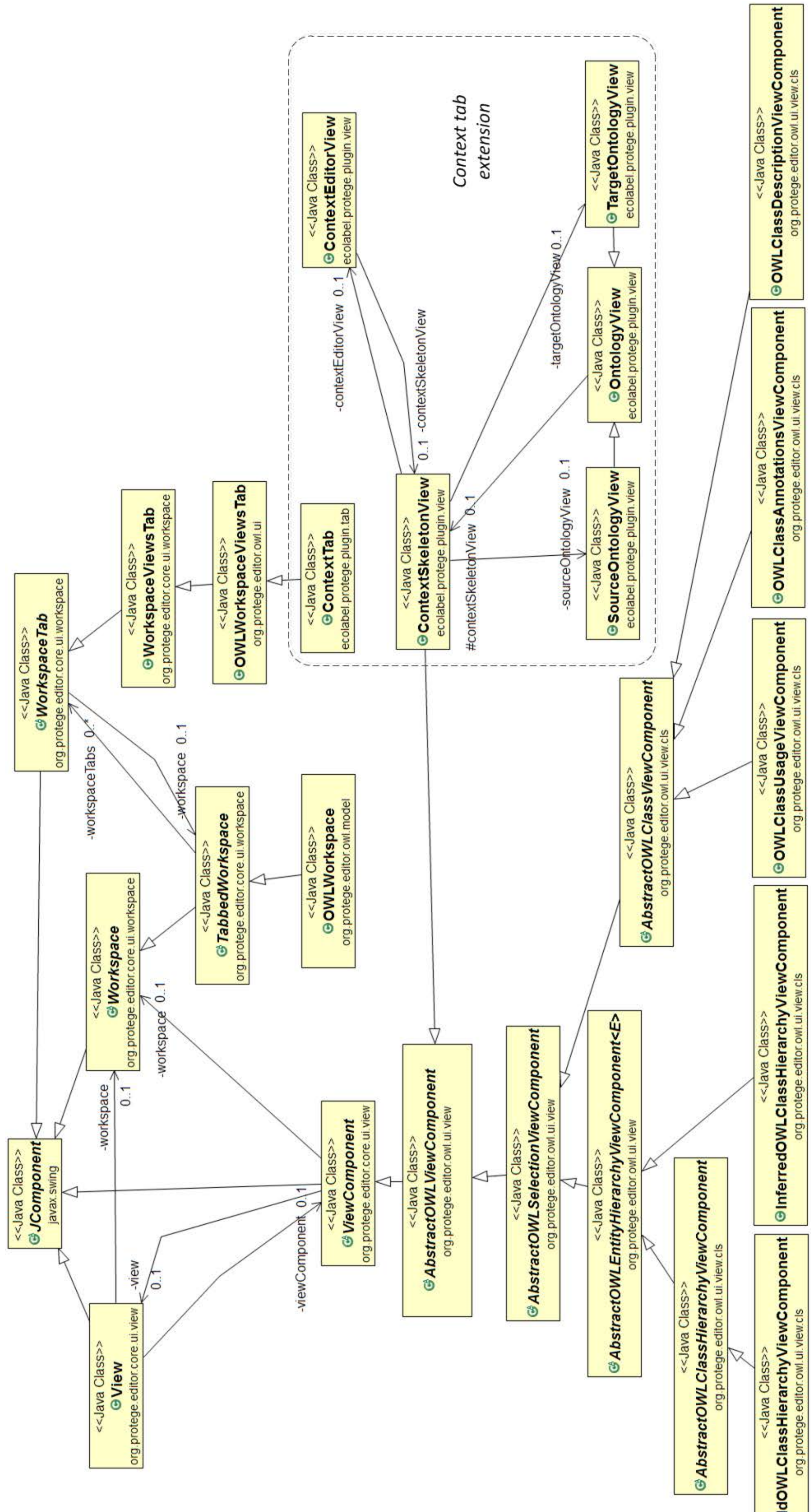


Figure 6.5: Our Context tab is an plug-in extension of the existent architecture.

6.2.3 Implementation

In order to implement this plug-in tool, Java [13] and Eclipse IDE [14] are used. Maven [15] is used to manage and build the project. We have applied OWL API [16]to implement most of the ontology related functions. Third party API JUNG[17] is used to implement the dependency graph of integrated ontology modules. Since this is not a technical report, we will not introduce too much trivial engineering details. A overall introduction about the program and some key implementation will be the main content of this section.

Figure 6.6 is a more detailed version of the Context tab extension in Figure 6.5. Due to the space limit, a full class diagram and sequence diagram explanation is not included here. In Figure 6.6, only main classes about the implementation and their attributes are listed. In the previous section, we have introduced that class *ContextEditorView* is the main editing area. Some visual components, e.g. top level menu, menu items as well as containers are defined inside. Each *ContextEditorView* instance is actually composed of an instantiation of *ContextHeaderComponent* and a set of *BinaryMappingBundleComponent*. Each *ContextEditorView* is also attached to an instance of *Context*. Here we assert that only one context configuration file can be opened and manipulated in one *ContextEditorView* and only one *Context* instance will be initialized. For each instance of *BinaryMappingBundleComponent*, multiple instances of *BinaryMappingComponent* will be contained, which means multiple mappings or relations can happen between any two ontology modules that are connected by a *BinaryMappingBundle*. *LocalOWLModule* class is used to store and manage the ontology integration. *ContextMemberFilter*, which is also part of *Context*, is responsible to maintain a checklist or filter for those selected entities in Context. *ContextVirtualVonsole* is responsible to print message in the GUI. *ContextChecker* will check and make sure that the contextual ontology integration is logic consistent at last.

As we have discussed in Chapter 4, our CIMOn method supports various OWL modeling expressions. Third party relations are also allowed. In the favor of scalability and extensibility, a combination of Strategy Pattern[18] and Simple Factory Pattern[19] is applied so that new OWL modeling expressions can be added easily. Figure 6.7 demonstrates how this combination works. For each OWL modeling expression, three basic functions are needed: i) parse the XML format expression into ontology axioms which is implemented by *parseImpl* (We have introduced in Chapter 4 that the context config file is in XML format, where cross module relation expressions and axioms are stored inside.); ii) render the expression in the GUI which is implemented in *renderMappingImpl*; iii) build OWL expression when user finishes editing and saves the context configuration file, which is implemented in *buildRealBinaryMappingImpl*. On the right side of Figure 6.7, these three functions have been

[13] JavaSE-1.8 (jre1.8.0_121)

[14] Eclipse Java EE IDE for Web Developers. Version: Mars.2 Release (4.5.2) Build id: 20160218-0600

[15] Apache Maven. Version: 3.3.3

[16] Version:4.2.8

[17] Java Universal Network/Graph Framework. Version: 2.0.1

[18] https://en.wikipedia.org/wiki/Strategy_pattern

[19] https://en.wikipedia.org/wiki/Factory_method_pattern

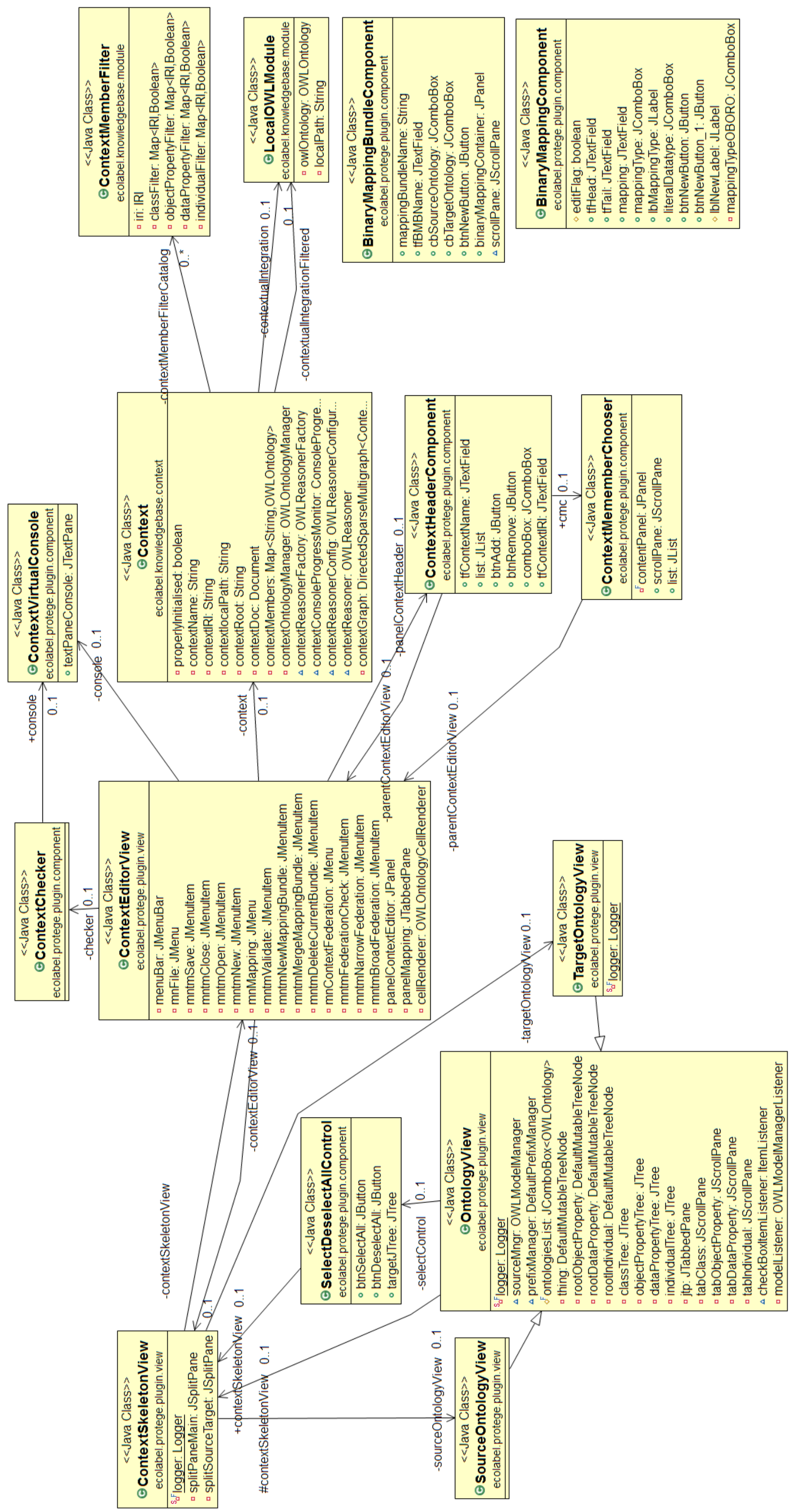


Figure 6.6: Main classes that build up the Context tab plug-in extension.

implemented in every class that is named with an OWL modeling expression. These classes have common super class *MappingOWLImpl*. Class *MappingOWLAdapter* is the only interface that the other part of the program is aware of. When it is necessary, plug-in program will initialize an instance of *MappingOWLAdapter* and pass a *String* parameter to it. Then *MappingOWLAdapter* will initialize different sub-class of *MappingOWLImpl* to realize different functions. Note that, for some third party relation expression, extra semantics may be needed. They can access these information from classes e.g. *SemanticCacheOBORO* which is in the left bottom of the figure. If new extension of OWL expression or any other third party relations are to be added, what we need to do is just adding a new implementing class inheriting *MappingOWLImpl*, and modifying the factory function in *MappingOWLAdapter*.

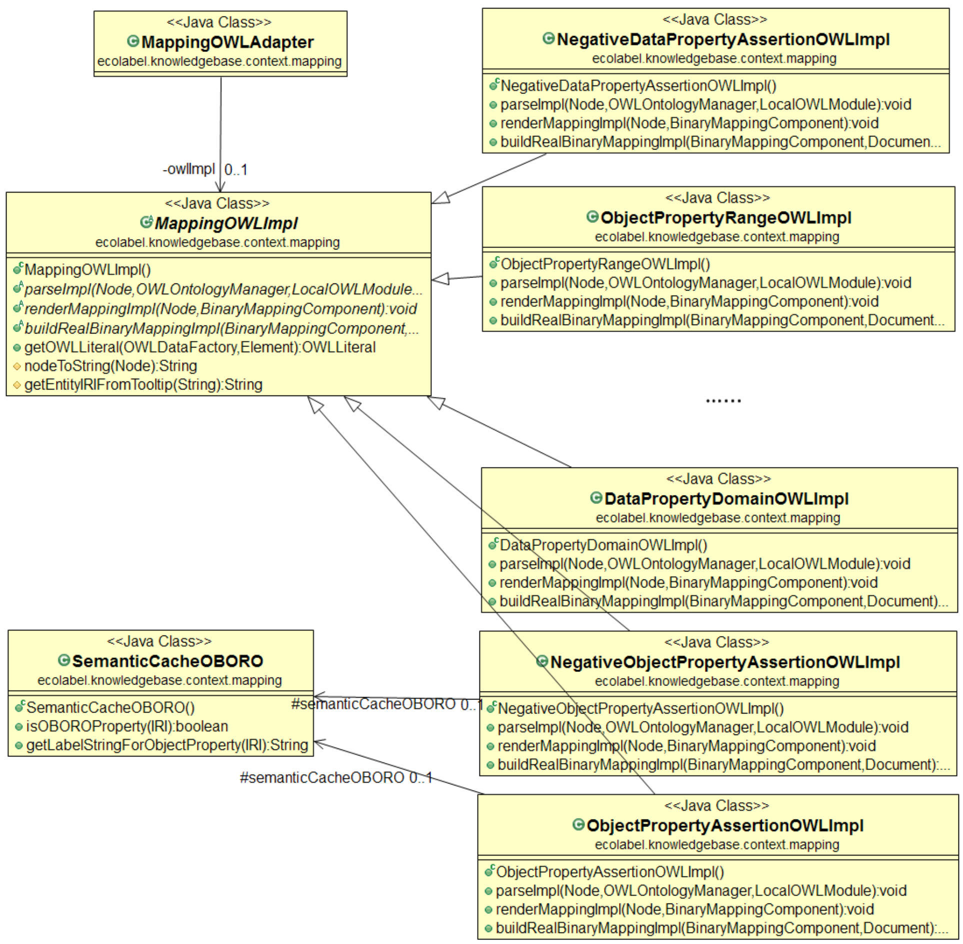


Figure 6.7: A combination of Strategy Pattern and Simple Factory Pattern is used to modularize the implementation of each cross-ontology relation.

Now, let's take a good look at the vivid GUI of the plug-in. Figure 6.8 is a preliminary implementation of our plug-in tool integrated in Protégé. By using this plug-in, we can edit the context of integrated ontologies as introduced in Chapter 4. This plug-in is a tab with name *Context tab*, user can find this under commend item Window → Tabs → Context tab. Of course, in order to let Protégé find the plug-in, a jar build must be ready under Protégé's plug-ins folder.

The Context tab is divided into three parts. The very left view is *SourceOntologyView*, the one next to it on the right is *TargetOntologyView*, and the main editing area on the rightmost is the *ContextEditorView*. We have formulated in Chapter 4 that, a pair of source ontology module and a target ontology module should be specified for each *BinaryMappingBundle*, so two ontology view components are set on the left of the GUI. On the top side of each ontology view component, a combo box is set to allow user to switch between opened ontologies in Protégé. Below this combo box selector, four tabs are set to browse different kinds of entities in the selected ontology. Here we don't need to rebuild a full set of ontology browsing functions as Protégé is already capable of basic ontology browse and manipulation. The ontology view in Context tab works mainly as an entity selector. If user wants to check entity details and related axioms, he or she can as well change to Protégé's original tabs. Each entity tab is equipped with *SelectAll* and *DeselectAll* button to facilitate user's selecting operation. In the front of each entity item in the hierarchy tree, a check-box is provided allowing user to select or deselect single entity.

On the right side of Figure 6.8, firstly, a menu bar is provided. User can create, open, save and close context configuration file. Mapping menu and its sub items allow user to add, delete, and validate *binaryMappingBundles* for each context. Contextual Integration and its sub-items will generate broad and narrow ontology Contextual Integration. A Contextual Integration of ontology is actually an ontology integration based on CIMOn. In Chapter 4, we have claimed that a Contextual Integration is more like a virtual ontology, here user can choose to materialize the Contextual Integration into concrete ontology files. Each Contextual Integration will be an ontology in OWL format. Broad Contextual Integration means all the entities and axioms of the context members should be integrated, even though those non-selected ones. Narrow Contextual Integration means only the selected entities and their axioms can be integrated. Broad federation provides more inference possibilities, more new axioms or knowledge may be inferred. Narrow federation tries to keep the integration result clean and as small as possible.

ContextEditorView is mainly divided into three parts (Figure 6.8), the upper part is *ContextHeaderComponent*, the middle part is a *TabbedPane* of *BinaryMappingBundleComponent* and the *Console* in the very lower part. Every time when a context is edited, context IRI, context name, context members as well as the root of the context must be specified. When the *Add* button beside the context members list is clicked, a dialog with list of all the opened ontologies in Protégé will show up, then user can choose the ontologies he needs as new context members (Figure 6.9). Or, user can directly click and choose the ontology modules that he does not want, then click the "Remove" button to remove the selected item from current context.

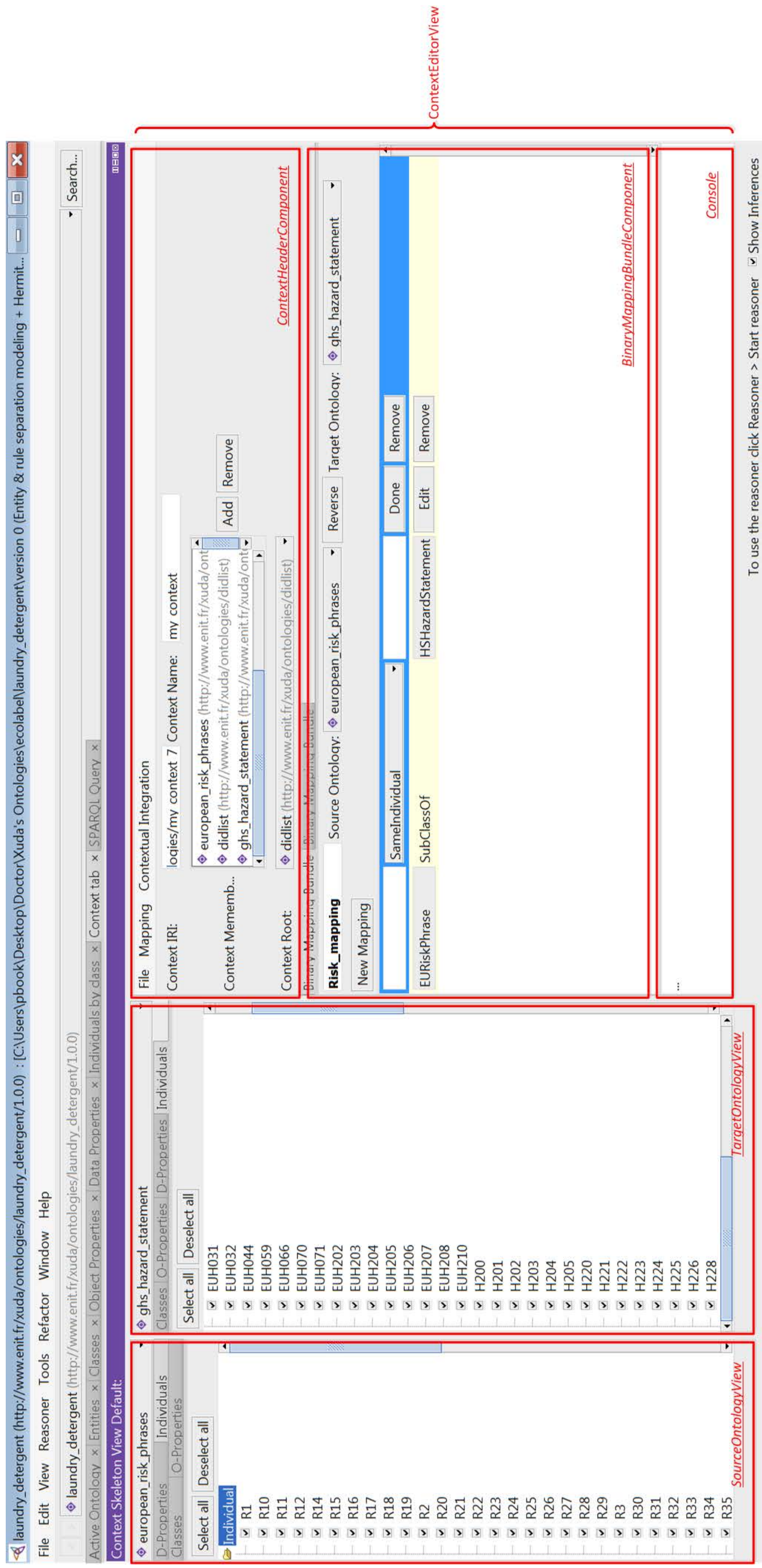


Figure 6.8: An overview of the Context tab plug-in for Protégé.

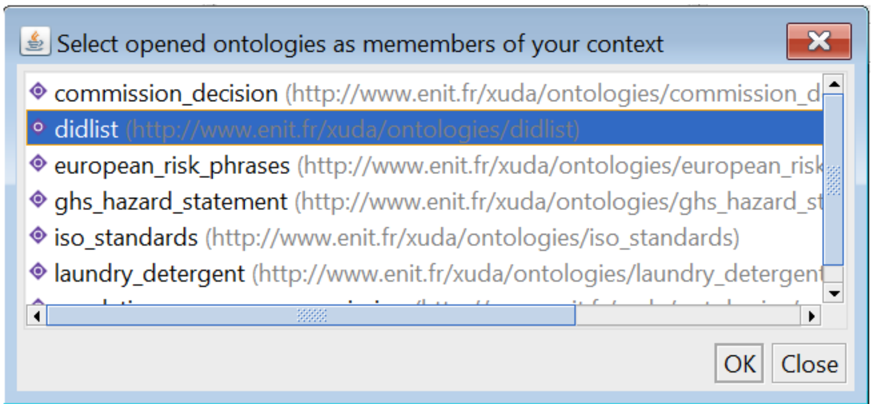


Figure 6.9: When the “Add” button on *ContextHeaderComponent* is clicked, user can choose the ontologies he needs from the list.

User can add a new *binaryMappingBundle* by clicking the command item “Mapping” → “New mapping bundle” on the *ContextHeaderComponent* (Figure 6.10). Each time a new *binaryMappingBundle* is added, a tab will be added on the *BinaryMappingBundleComponent*. Commend item “Mapping” → “Delete current mapping bundle” can remove currently editing bundle tab and delete it from the context. For each *binaryMappingBundle*, a user defined name can be applied. Also, the directionality of the bundle must be specified. Combo-box of context members are provided to let user choose the source ontology and the target ontology. When the *Reverse* button is clicked, the directionality will be reversed.

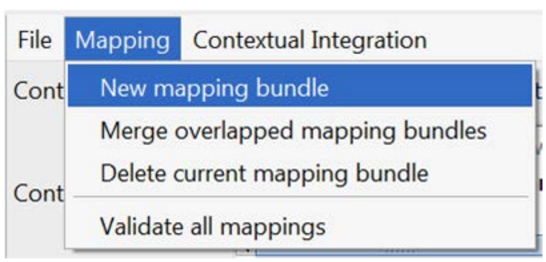


Figure 6.10: *Mapping* command and its sub-items.

Within each *BinaryMappingBundleComponent*, a list of *BinaryMappingComponent* is rendered according to different OWL expression and relations (Figure 6.11). When the *New Mapping* button on each *BinaryMappingBundleComponent* tab is clicked, a new relation item will be added in the *ScrollPanel* below. When the *Remove* button at the end of relation item is clicked, this item will be correspondingly deleted. When the *Edit* button is clicked, the item's background color will be changed and it will enter the editing mode. User can select and drag entities (class, object property, data property, and individual) from hierarchy tree on the left side to subject position or object position to edit current relation item. Combo-box of relations is also provided to switch between the supported OWL expressions or relations. For the entities either rendering in *SourceOntologyView* or *TargetOntologyView*, a check-box is provided allowing user to decide whether to keep this entity in the context that is being edited or not. In the case of Figure 6.11, we can see that several individuals have been chosen.

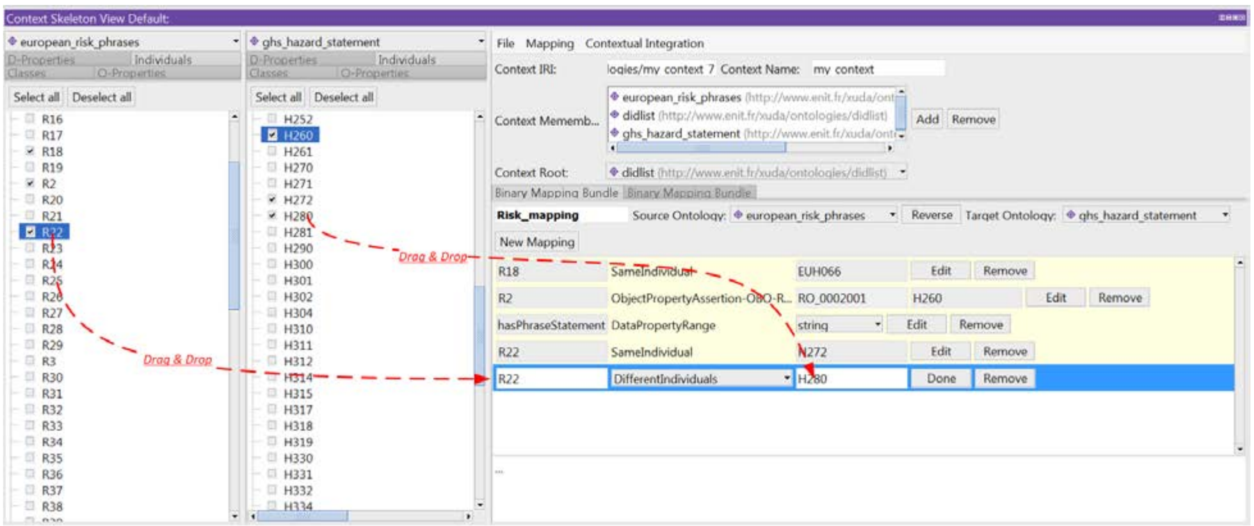


Figure 6.11: User can directly drag and drop the entity he needs to each *BinaryMapping* item that is being edited.

When all the editing for a context is done, user can save the context by clicking "File" → "Save" (Figure 6.12). Before saving the context, a Contextual Integration check is suggested as it is very important to keep the context editing result i.e. Contextual Integration consistent. When the Contextual Integration check is launched (Figure 6.13), Hermit reasoner will take account all the selected entities, put them together, and do the reasoning. The check result will be output in the console at the very bottom of the GUI.

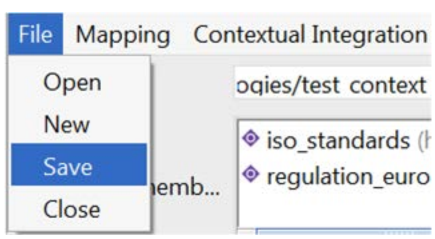


Figure 6.12: *File* command and its sub-items.

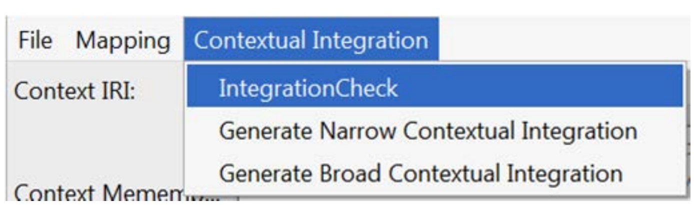


Figure 6.13: *Contextual Integration* command and its sub-items.

When this manuscript is being written, we used Metrics[20] plug-in for Eclipse to do a basic code analysis on our source code. The quality of the code is acceptable. Figure 6.14 is the snapshot of the report generated by Metrics. Most metrics of our code conform the criteria of the plug-in except for McCabe Cyclomatic Complexity and Nested Block Depth. We will keep working on this development and improve the program's functionality and quality. A build of this plug-in can be found on GitHub: https://github.com/xudaddd/CIMOn. We will publish the source, tutorial and other documentation as soon as possible.

Metric	Tot...	Mean	Std. Dev.	Maximum
Number of Parameters (avg/max per method)		1,028	1,228	7
Number of Static Attributes (avg/max per type)	8	0,136	0,342	1
Efferent Coupling (avg/max per packageFragment)		4,25	4,952	18
Specialization Index (avg/max per type)		0,22	0,657	3
Number of Classes (avg/max per packageFragment)	59	4,917	5,837	18
Number of Attributes (avg/max per type)	175	2,966	5,307	23
Abstractness (avg/max per packageFragment)		0,018	0,047	0,167
Normalized Distance (avg/max per packageFragment)		0,379	0,312	0,92
Number of Static Methods (avg/max per type)	3	0,051	0,22	1
Number of Interfaces (avg/max per packageFragment)	0	0	0	0
Total Lines of Code	5338			
Weighted methods per Class (avg/max per type)	635	10,763	14,668	67
Number of Methods (avg/max per type)	321	5,441	8,041	51
Depth of Inheritance Tree (avg/max per type)		2,949	2,004	7
Number of Packages	12			
Instability (avg/max per packageFragment)		0,603	0,302	1
McCabe Cyclomatic Complexity (avg/max per method)		1,96	2,859	22
Nested Block Depth (avg/max per method)		1,386	0,951	8
Lack of Cohesion of Methods (avg/max per type)		0,255	0,384	1,214
Method Lines of Code (avg/max per method)	3350	10,34	22,939	225
Number of Overridden Methods (avg/max per type)	13	0,22	0,554	3
Afferent Coupling (avg/max per packageFragment)		5,833	8,572	26
Number of Children (avg/max per type)	22	0,373	1,965	15

Figure 6.14: A code analysis report generated by Metrics plug-in in Eclipse.

6.3 Decision support system prototype for EU Eco-labeled laundry detergent product

In this section, we will present the implementation of the decision support process proposed in the previous chapter, i.e. the decision support system prototype.

[20]http://metrics.sourceforge.net/

6.3.1 About EU Eco-labeling toolkit

First of all, let's have a look at how the current labeling process for laundry detergent product works. EU Eco-label has been undergoing for more than twenty years as the only official eco-label in the whole European Union, a well-defined coordination between the EU Commission and other member countries' competent bodies has established. On the official site of EU Eco-label[21], detailed documentation is provided to enterprises to facilitate the application process. On the same site, there is also a very detailed product group catalog and the corresponding criteria for each product or service group.

Before we introduce the decision support system prototype. Current EU Eco-labeling toolkit will be presented. To start an application, it is suggested by EU Eco-label that an on-line application management tool called ECAT_Admin[22] should be used. Through ECAT_Admin portal (index page in Figure 6.15), detailed description of the applicant enterprise and the target product or service will be required. However, the information gathered here is no more than some basic registry information. This on-line tool will facilitate the office work for application management, but not the pivotal product evaluation part. As for the evaluation, detailed parameters, description as well as test reports about the product or service are needed, while, we don't see this part in this ECAT_Admin tool.

[21] http://ec.europa.eu/environment/ecolabel
[22] https://webgate.ec.europa.eu/ecat_admin/

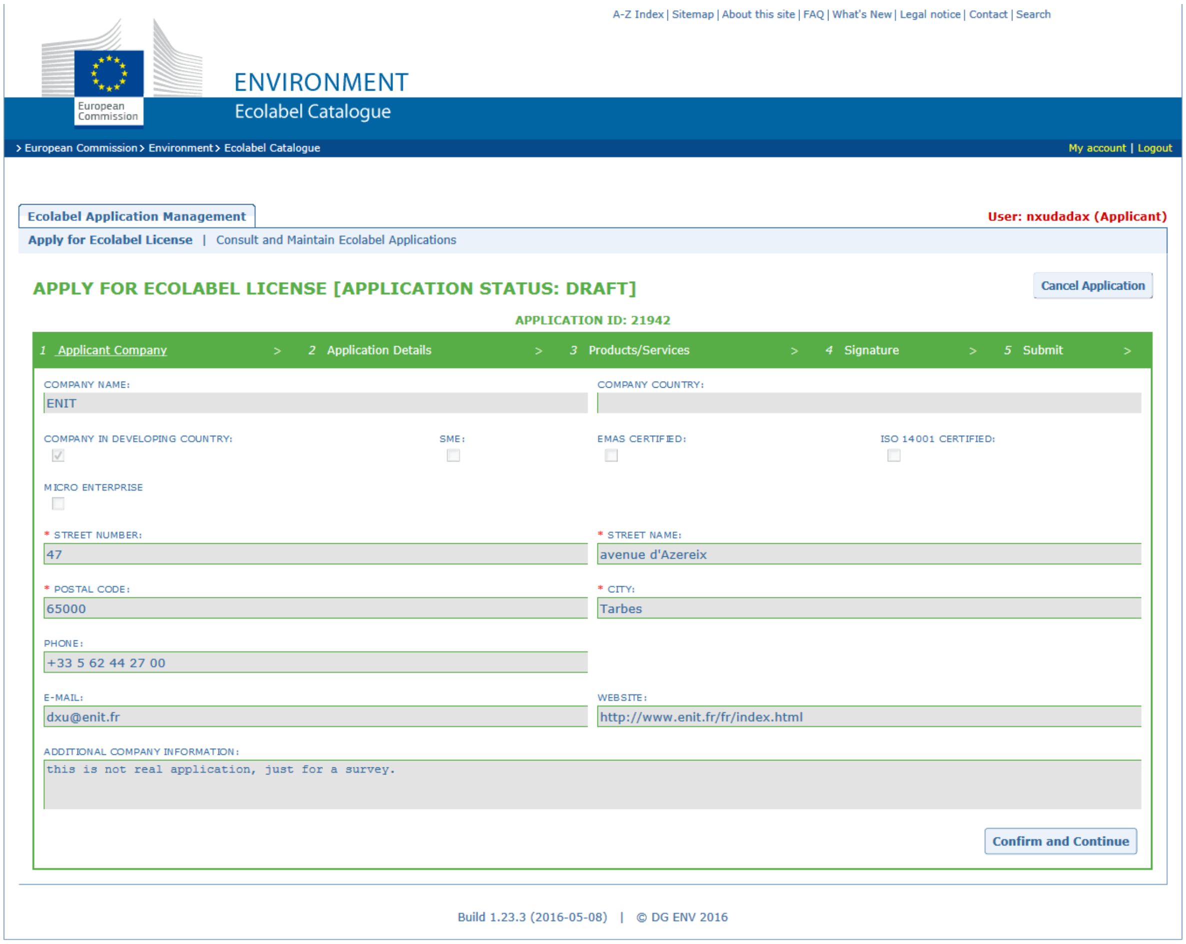


Figure 6.15: EU Ecolabel online application tool, ECAT_Admin. Brief registry information is required.

On the other hand, as quoted from the instruction: "Once you have submitted the online application, you will need to submit the required paper file to your Competent Body. Within two months of your initial application submission, your Competent Body will assess your product against the criteria set for it."[23] Here the assessment process is an important aspect of what our research is about. As indicated by the user manual and check-list of each product or service group, an "application pack" composed of various files and forms is required by the competent body. Sometimes, extra excel files or other calculating sheets are required too. In the end, to handle and validate all the information recorded in such a series of files and forms, it will be a tiring work for human experts from competent body. Figure 6.16 is one of the required form in the application pack of laundry detergent product. Applicant has to fill similar forms and send them to the competent body. These traditional forms and excel calculating sheets are difficult to be managed actually. We have introduced a little bit about the evaluation of some criteria in Chapter 3. Some mathematical formula must be used to calculate some parameters e.g. Critical Dilution Volume. Here in Figure 6.17, an excel sheet

[23] http://ec.europa.eu/environment/ecolabel/how-to-apply-for-eu-ecolabel.html

is provided in application pack for applicant to generate such parameters. Applicant must download, complete the excel file and then send it to the Competent Body along with other documents together. In our opinion, this information collection is inefficient and unsafe. Although some tools have been used to help human experts, it is not DSS. As data is stored and managed in a non-collective way, data query and analysis are very difficult to be carried out. In our research, we will try to digitize such information collection method, store the information in ontology knowledge base and more importantly, automatize the evaluation process in our decision support system.

2 PRODUCT FORMULATION

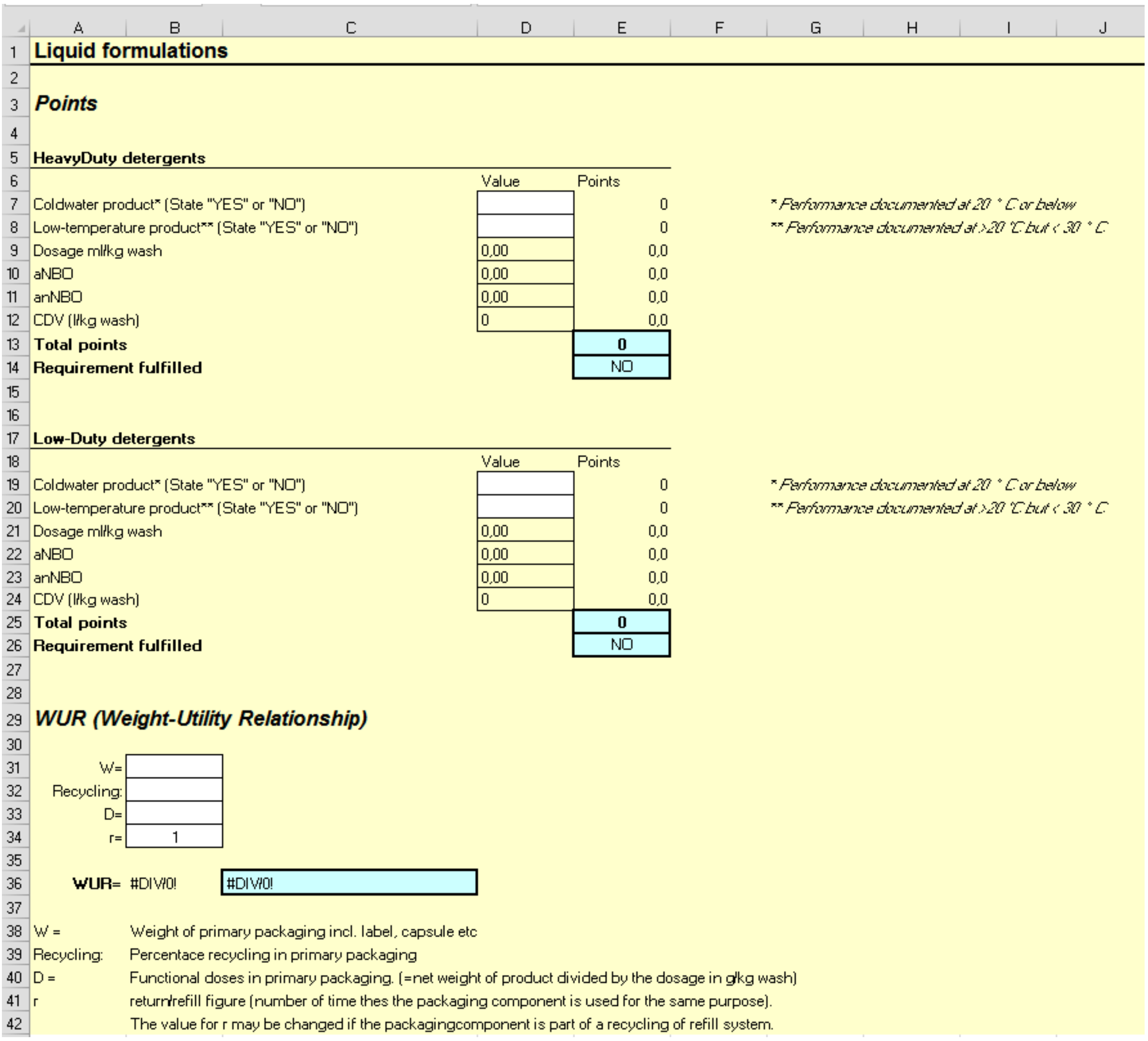
The ingredients and the water content of the candidate product shall be listed as illustrated by the table below. If an ingredient (except for fragrance mixtures) consists of more than one chemical compound, all individual compounds must be listed and the content specified (in % of the product). A safety data-sheet (SDS) shall be enclosed for all raw materials in the product. The concentration of ingredients in the product, which implies a requirement for documentation of compliance with the ecological criteria, is generally defined at ≥0.010% by weight of the preparation. For preservatives, colouring agents and fragrance mixtures, compliance with the ecological criteria is required regardless of the concentration unless otherwise specified.

MANUFACTURER'S DECLARATION

Water content of the candidate product: …..................... % (w/w)

Substance		Function in product (e.g. surfactant, builder, preservative)	CAS No. (or CI No. or other precise description)	DID Number (if applicable)	Concentration (%, w/w)	SDS, Appendix No.
Trade name	Chemical name					

Figure 6.16: Application form example: product formulation and ingredients declaration.



	A	B	C	D	E	F	G
1	**Liquid formulations**						
2							
3	***Points***						
4							
5	**HeavyDuty detergents**						
6				Value	Points		
7	Coldwater product* (State "YES" or "NO")				0		* Performance documented at 20 ° C or below
8	Low-temperature product** (State "YES" or "NO")				0		** Performance documented at >20 °C but < 30 ° C
9	Dosage ml/kg wash			0,00	0,0		
10	aNBO			0,00	0,0		
11	anNBO			0,00	0,0		
12	CDV (l/kg wash)			0	0,0		
13	**Total points**				**0**		
14	**Requirement fulfilled**				NO		
15							
16							
17	**Low-Duty detergents**						
18				Value	Points		
19	Coldwater product* (State "YES" or "NO")				0		* Performance documented at 20 ° C or below
20	Low-temperature product** (State "YES" or "NO")				0		** Performance documented at >20 °C but < 30 ° C
21	Dosage ml/kg wash			0,00	0,0		
22	aNBO			0,00	0,0		
23	anNBO			0,00	0,0		
24	CDV (l/kg wash)			0	0,0		
25	**Total points**				**0**		
26	**Requirement fulfilled**				NO		
27							
28							
29	***WUR (Weight-Utility Relationship)***						
30							
31	W=						
32	Recycling:						
33	D=						
34	r=	1					
35							
36	**WUR=**	#DIV/0!	#DIV/0!				
37							
38	W =	Weight of primary packaging incl. label, capsule etc					
39	Recycling:	Percentace recycling in primary packaging					
40	D =	Functional doses in primary packaging. (=net weight of product divided by the dosage in g/kg wash)					
41	r	return/refill figure (number of time thes the packaging component is used for the same purpose).					
42		The value for r may be changed if the packagingcomponent is part of a recycling of refill system.					

Figure 6.17: To generate detergent product's parameters in the application form, extra calculating excel sheet is provided.

6.3.2 System design

Java Swing API and OWL API are used to build this decision support system prototype. We introduce in Figure 6.18 the class diagram including the main classes of the system. We will use this diagram to explain our basic system design.

The basic design spirit of our system is quite simple actually. It follows most of the processes that we have introduced in Chapter 5. In the class diagram, we have 3 class hierarchies whose root class are *ApplicationOntologyBuilder*, *Application*, and *ApplicationOntology*. Class *Application* is used to carry product or service's detailed information. *ApplicationOntology* is an ontology format of *Application*. We have explained in Chapter 5 that we need to retrieve and transform product profile into ontology. *ApplicationOntologyBuilder* is responsible to commend and realize such transformation by using *Application*, and then the result of this transformation is *ApplicationOntology*. Once we get the result in *ApplicationOntology*, we can proceed to the following reasoning process. Rules from criteria ontology will be applied in the instance of *ApplicationOntology* and reasoner will be started. We can see that the structure of these three class hierarchies is quite similar. The sub-classes of *ProductApplication* should be the different product groups defined in the EU Eco-label catalog. While, we have for now only implemented laundry detergent product group as single study case. We divide different kinds of requested product's information into different *Action*. Another layer of abstraction called *ActionHub* is added as member attribute of concrete *ProductApplication* (LaundryDetergentProductApplication in our case).

6.3.3 Implementation

The main interface of this prototype decision support system is illustrated in Figure 6.19. This prototype software serves to the domain experts and decision makers who work for the eco-labeling. Based on this prototype, we can also develop some application assistance tools for the applicant. Before the applicant submits their product or service profile, he can use such kind of tool to do some preparatory test and verification. In this prototype system, we have each application stored in a file of certain format (Standard XML has been realized). The user loads the application file and then a tree structure will show up on the left side to allow the user to navigate between parts of the application. When an item in the tree navigator is clicked, the main panel on the right will change to corresponding page for user editing or review.

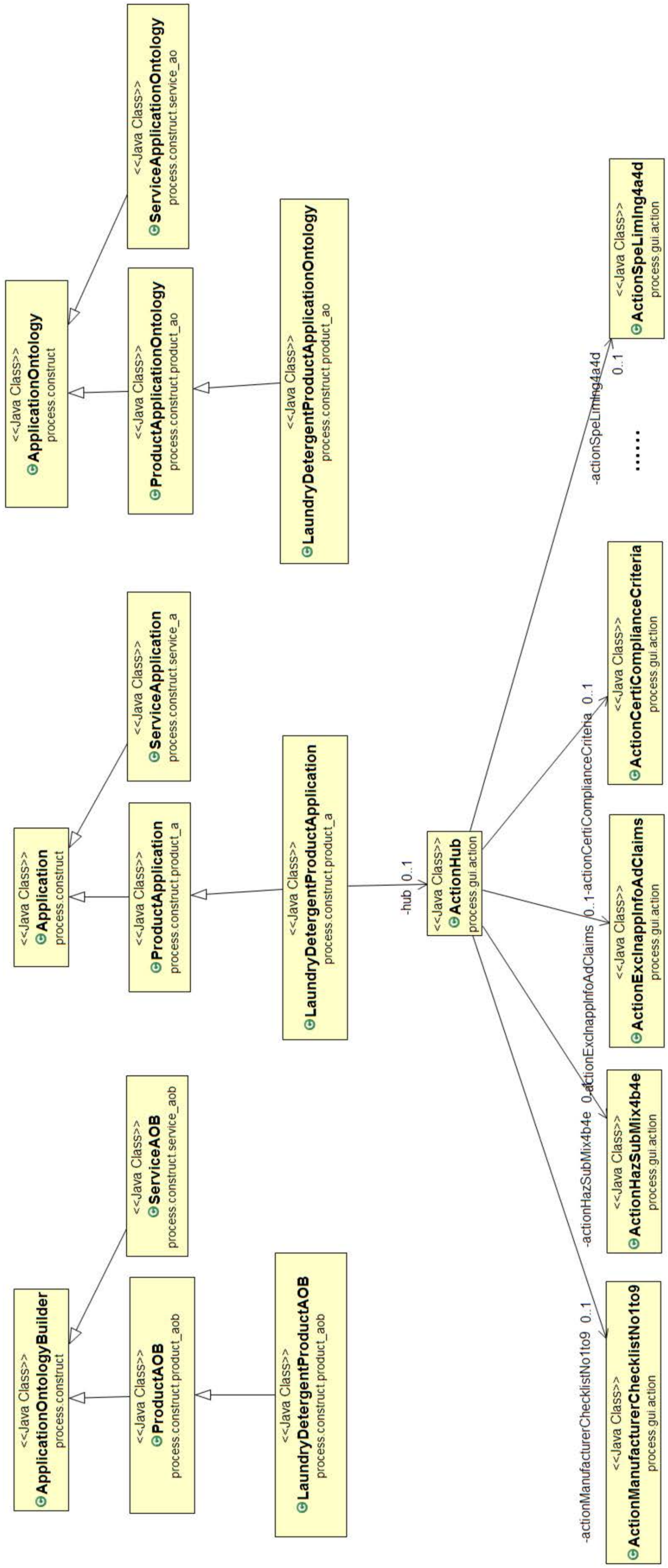


Figure 6.18: Decision support system main class diagram

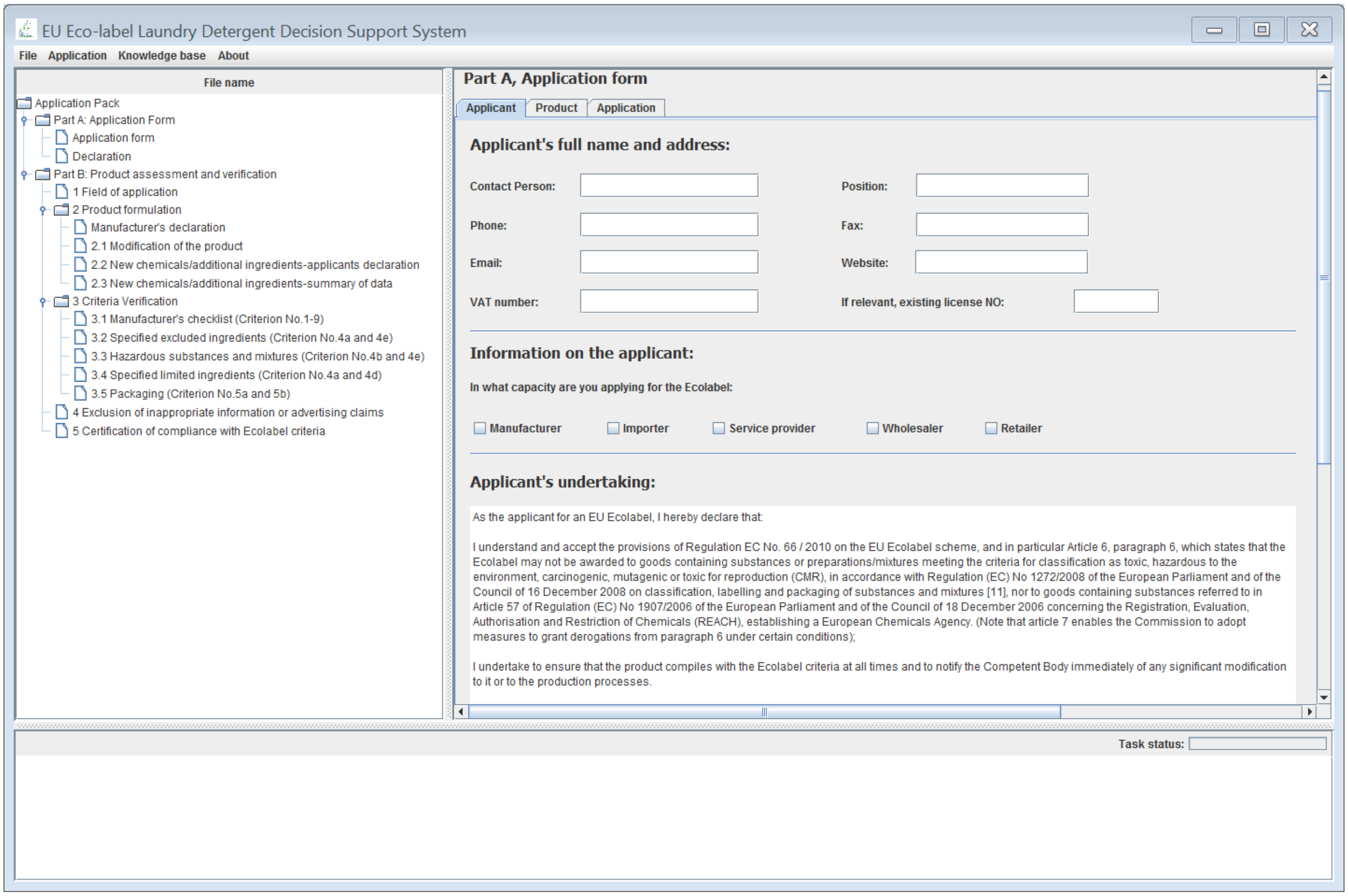


Figure 6.19: A Java prototype system for the eco-labeling decision support platform.

When the user thinks that the editing is OK, he or she can use commend "Application" → "Check" to check the completeness of the profile. Then commend "Application" → "Start EU Eco-labeling process" will start the evaluation. Before the eco-labeling decision support process is started, the system will retrieve the product's profile (Class *Application*) from the application file, combine it with the corresponding product Template Ontology to build a Product Profile Ontology (Class *ApplicationOntology*). Then, reasoning based on the criteria rules will proceed and the result will be shown in the console panel at the bottom of the GUI. In Figure 6.20, reasoning result of a laundry detergent product case study is presented. The text in blue record all the explanations we can get from system with the help of reasoner. The first line of each explanation block is the conclusion of current reasoning result. In the case of Figure 6.20, we can learn that, the product with ID "low_duty_laundry_detergent_example_0" is rejected, i.e. it should not be eco-labeled. In the rest part of the explanation, the violation of certain rules can be found. Experts can make decision based on these reasoning result and explanations. Normally, if negative reasoning result appears, rejective decision should be made, then human experts can inspect the product's profile and feedback to the applicant with proper argumentations. (In our prototype system, the default product Template Ontology is the laundry detergent. Based on the CIMOn method that we introduced in Chapter 4, the layered architecture should be implemented between decision support system and the ontology knowledge base in order to allow the application to have more ontology access options. However, we could not finish this due to the time and resource limit. We will discuss this more in the last Chapter.)

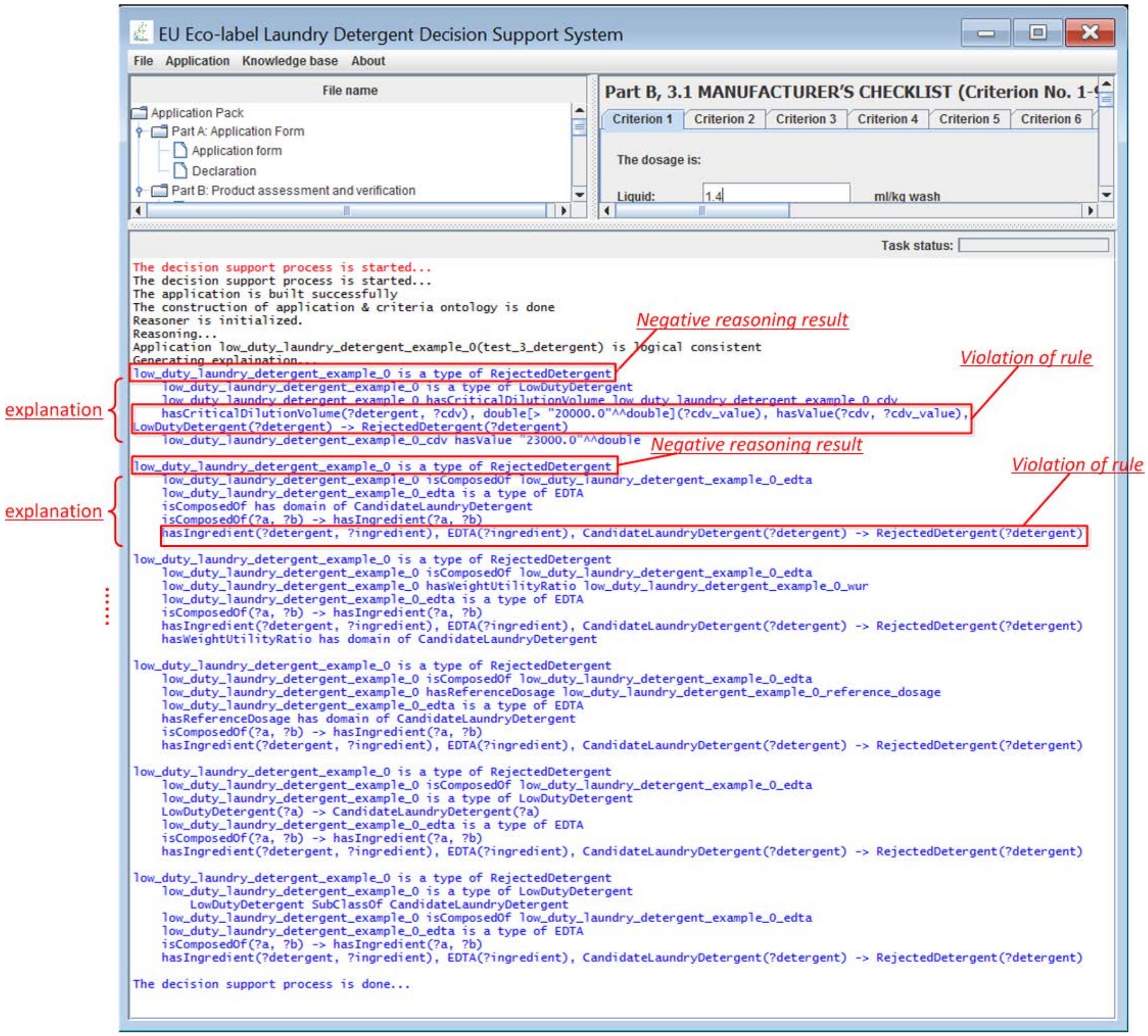


Figure 6.20: How to interpret the reasoning result and make decision.

The prototype system also provides a query portal which allows users or other applications to launch standard SPARQL query to access the ontology knowledge base. From the perspective of data sharing and reuse, our system is more like a platform rather than a decision support dedicating system. This function is implemented with the Jena API[24], we are trying to publish this query portal on the web. Figure 6.21 is the interface for this query portal and the query result will be displayed in the console panel.

[24] https://jena.apache.org/

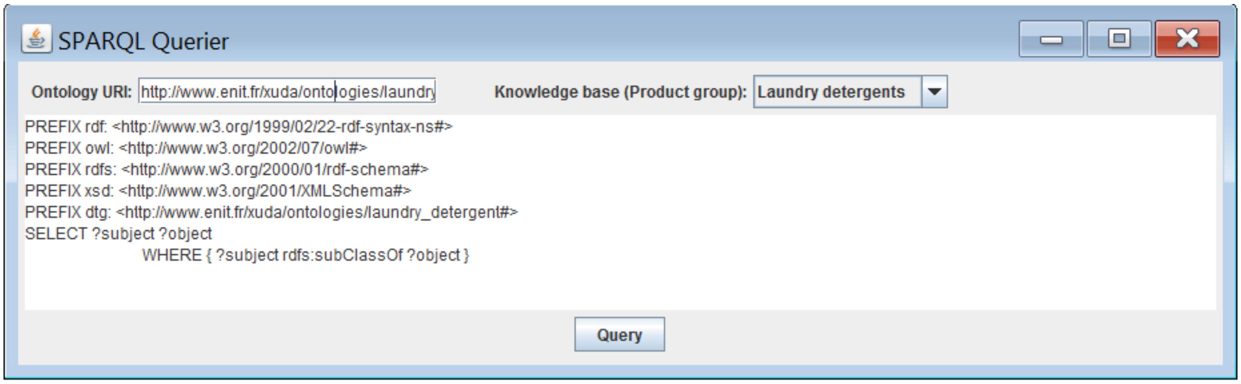


Figure 6.21: A SPARQL query portal of the prototype system.

6.4 Conclusion

In this last chapter, we have demonstrated a plug-in tool for the Contextual Integration for Modular Ontology (CIMOn) method which was introduced in Chapter 4. We have also presented more implementation related aspects about the eco-labeling decision support system. The contribution of this chapter is that we presented the concrete materialization of our theory and method which were elaborated previously. Our theory and method have been validated. We hope that our practice and experience could be inspiring and useful to other researchers when they seek to implement similar tasks or systems. After this chapter, we will have a summary of the whole thesis; the limitation and future work of our research will be discussed as well.

Chapter 7

Conclusion, discussion, limitation and future work

7.1 Summary and conclusion

To democratize eco-labeled products and services in order to achieve a more green and sustainable economics, a better eco-labeling process is needed. In this work, we've introduced what is eco-labeling and EU Eco-label, their main concern, objective and challenges (Chapter 1). The main objective of our research is to improve eco-label and eco-labeling process in order to popularize eco-labeled products and services so as to achieve a more competent and ecological economy. To realize such objective, the basic solution is a decision support process based on an ontology knowledge base (Chapter 5). In order to do this, various domain ontologies and modules of ontology or rules have been stored in this knowledge base (Chapter 3). Also, we have proposed a Contextual Integration for Modular Ontology (CIMOn) method to manage and integrate these ontology modules (Chapter 4). At last, we have developed and validated the proposed decision support system prototype and context editor plug-in (Chapter 6). At least three main contributions can be identified in our work: (i) modularization design and development of the ontology knowledge base. This work gives our answer to the questions that we have asked in the introduction chapter, i.e. how to achieve appropriate eco-labeling knowledge representation with ontology and how to handle the evolution of eco-labeling criteria; (ii) proposition and prototype implementation of the decision support process, which answers how to define and formalize a computerized decision support process for Eco-labeling; (iii) CIMOn and its plug-in implementation in Protégé editor. This contribution has answered the question how to better reuse and integrate ontology modules, etc.

7.2 Discussion

Although we've built an ontology knowledge base for EU Eco-labeling laundry detergent product, we should not exclusively burden all the work on the knowledge base. The criteria document we have used as knowledge base reference is published in *Commission decision of 28 April 2011 on establishing the ecological criteria for the award of the EU Ecolabel for laundry detergent 2011/264/EU.*[1] This commission decision document is composed of regulation articles, annex where each item of the criteria is explained, as well as appendix. The regulation articles are not very interesting as it gives only administrative declaration and reference. Most of the knowledge about laundry detergent is elaborated in the annex and appendix. Among these criterion, not all of them are suitable and easy to be translated into ontologies. Part of this is due to the limit of expressiveness of OWL language and SWRL rules. Another reason is because sometimes well-informed and experienced human labor seems more competent for aesthetic and usability assessment. For example, the criterion NO.8 of this laundry detergent criteria talks about the consumer information such as the dosage instructions, information on the packaging, and additional claims on the packaging. Instead of assigning these work to human experts, we can imagine how hard and costly it would be to implement and train an AI system to do that. For the sake of a better performance of the decision support process, we propose to take a trade-off strategy that part of the criteria inspection will still be implemented by traditional program logic. Thus, our decision support system will a good tool to assist and accelerate the eco-labeling process. Human experts can be liberated from heavy document reading and assessment work and spare more time on the work that really need sophisticated experience and inspection at scene.

We think that modular ontology or ontology modularization is a promising approach to realize large scale application of ontology. Analogous experience is learned in many other engineering domains, i.e. the more specific and unitary function one artifact has, more often it could be reused. The screw, no matter who the manufacture is, as long as it follows certain standard in format and material. it can be replaced anywhere. If we inspect the microscopic world of nature, we can find the same pattern. For instance, water molecules on earth are of the same composition and the same chemical and physical characteristics, no matter the ones you drink at lunch today or those circulating in the brook behind your house. They are the same and can be reused everywhere. We live in a world of modularization. In today's software engineering, especially in open source community, large amount of ready-made programing libraries and middle wares are available. In the community of knowledge engineering, people are trying to do the same thing, more and more institutes and researchers have published their ontologies on web. By the help of OWL, RDF and other semantic web standard, faster knowledge sharing and search is becoming easier, and more powerful and intelligent applications are breeding.

It has been introduced in Chapter 3 that we tried to build ontology by using ontology leaning technologies. However, current ontology learning tools are not good enough to gen-

[1] http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32011D0264

erate satisfactory result. Limited number of terms or concepts can be identified. Identified properties are even fewer. Also, it seems very difficult to distinguish object property from data property, as well as identify proper domain and range concept by means of current ontology learning technologies, not to mention even more complex axioms. In fact, the same ontology learning technologies can be also applied to the product profile construction process that we described in Chapter 5. It will be indeed an appealing vision that we let the computers scan product profile document, then the content of the document would be understood well, at last the key parameters could be extracted and product profile is constructed. According to our experience, building ontology is not an easy job at all. Even reading the document could be quite annoying too. In that sense, to our knowledge, Natural Language Processing and related technologies will be quite useful in this context. Today, powerful QA systems have been implemented, e.g. IBM's Watson. How to utilize available NLP related technologies to improve ontology learning or ontology extraction will be very interesting and useful research topic.

In the end of our research, an ontology knowledge base and decision support system was proposed. Imagine that, as the labeling applications accumulate, historical application data set which contains large amount of product profiles would become available. How to dig more value from the knowledge base and data? To our knowledge, some statistics or machine learning technologies can be very helpful. How to transform ontology and semantic data into schemes that are applicable for machine learning technologies? How to apply current machine learning model and algorithms to ontology? Issues like this will also be worthy of research. Also, in order to evolve from Internet to the Semantic Web, enormous data and textual content has to be semantically labeled. From our perspectives, the combination of ontology, NLP, and machine learning related technologies could provide promising solutions for Semantic Web.

Also, privacy and security are very important issues. How to make use of ontology and semantic data while ensuring appropriate privacy policies and security of confidential industrial data? Can traditional information security technologies be adapted in ontology and semantic web related scenarios? When ontologies are reused by others, some kind of access control mechanism or protocol may be necessary.

Our work is about eco-labeling certification, other kinds of certification alike domains may have similar problems as ours. So, our ontology based decision support system could be applied to tasks like configuration check, simulation and test as well as other certification process.

7.3 Limits and future work

For the time being, the development of the DSS system and the refinement of the ontologies are undergoing. In the next phase of work, we will focus on more practical research and development. First, we plan to improve the argumentation generation. A more user-friendly explanation and argumentation report should be generated and Manchester Syntax should be ultimately translated into human language. Then, more product cases should be collected and analyzed. After statistical analysis and learning, we expect our decision support system can give improvement advice to those products that fail the eco-labeling assessment.

Our current research invests much efforts on domain-centric ontology knowledge base and ontology modularization, however, there is still a lot of work to do in order to finally achieve a convenient semantic interoperability with other external ontology repositories and other data sources. Re-usability is also a very important issue. For example, in our knowledge base for laundry detergent, an ontology module called Detergent Ingredient Database list (DIDlist) was developed. This module consists of hundreds popular ingredient chemicals for detergents and other similar products. In fact, we've found several other EU Eco-labeling product groups that use the same piece of ingredient database and that means the same DIDlist could be reused in the ontology knowledge base development of these products' criteria. In our next phase of research, we'd like to enrich our knowledge base by developing new product groups of cosmetics product, all-purpose cleaner and etc. Then, based on the local reuse across domains, external reuse and mapping are expected. Our knowledge base can be linked to other knowledge base and that is actually more relevant to ontology mapping or ontology alignment which will be an important topic in our future research work. Once the link and mapping between ontology concepts are established, further problems like distributed reasoning cross domains are also worth of research. So, in the next phase of our research, we are going to emphasize on these problematics.

In our research, only one product group was tested, it would be interesting to apply our methodologies to other sibling products. Another main limit of our research is the lack of more validation. We have only implemented some prototypes, which are not yet verified by other researchers and common users. Propositional solutions have to be verified and validated by practice. In the future, we plan to improve the development of the plug-in and the DSS prototype. The proposed CIMOn method only allows for user's customized entity selection. Since some module extraction algorithms and technologies are already available, we can provide such functions as extra options. As we can see in our CIMOn method, the alignment or relation editing is still manual, which is quite labor intensive process. Since quite amount of automatic or semi-automatic ontology alignment approaches have been proposed and implemented in the literature, we plan to include these technologies in our plug-in in order to facilitate and accelerate the cross-module relation editing.

According to the official description of EU Eco-labeling [2], even after the certification, a regular "follow-up" will proceed. "In order to guarantee consumers or users of your certified products or services that you continue to conform with the requirements of the 'NF Environment' or EU Eco-label marking schemes, AFNOR Certification will collect samples in your stores or factory workshops for product testing; it will also conduct regular factory workshop audits." Our work for now has not taken this follow-up into consideration. In the future work, a history of the product's profile as well as its samples that would be collected

[2]http://www.ecolabels.fr/en/layout/set/print/professionals-area/industrials-service-providers-eco-labels/the-5-steps-to-follow-to-obtain-certification

afterwards should be maintained. Analysis based on a series of profiles concerning the same product in different historical periods could be useful for decision makers. This is an issue that is worthy of study in our future work.

For now, the CIMOn method tries to follow the conservative extension principle that newly formed integration will not alter the already integrated module. However, we could not give logic formalism and proof based on Description Logic. We will put more focus on such issues in our future work. On the other hand, situations that violate this conservative extension principle may happen. To explain these, we will first check situation with conservative extension. For example, concept A and B are from ontology module X, an external relation r associate these two concepts via axiom α in the context. We assert that this kind of integration is conservative in terms of the signature(A, B), because the context only provide new semantics (axiom α) to the original module and the original semantics of A and B in module X is not changed at all. Even if some relation associate two concepts from different module, we can prove that the context is still conservative extension. While, let's check such a situation that, still in a context integrating module X, an axiom like this is introduced in the context, β: A *subClassOf* B. Now the integration based on this context is no more conservative extension any more. Because for the same signature(A, B), axiom β does not hold in the original module X, which means the new integration by means of the context is has changed the module X, then such extension is not conservative any more. However, this nonconservative situation may indeed happen when we want to add or alter the integrated module. (The proposition 35 in [110] indicates that E-Connections are not conservative.) In nonconservative situation, potential risks with respect to logic conflict and decidability problems exist. It is already known that determining whether an ontology safely extends (is a conservative extension of) another ontology is not decidable for expressive Description Logics such as OWL-DL [125]. In our future work of research, we will first try to find a decidable and efficient mechanism for CIMOn to generate conservative extension integration or approximation, then explore and extend CIMOn method to nonconservative extension integration.

As for the reasoning, what we have applied in CIMOn method is still an integral approach. Reasoning algorithm is applied and run on singular calculation unit. This approach could be a bottle neck when the ontologies are distributed or the integrated ontology is very big. Distributed or paralleled reasoning solution must be helpful to address such problems. We consider doing research on distributed reasoning and incremental reasoning to improve reasoning performance in terms of the contextual ontology integration. Although CIMOn has been successfully applied in local ontology reuse and integration scenario, it does not support distributed ontology reuse or integration via the web. Lack of heterogeneous data and syntax support is another limit as CIMOn has been only verified and applied on OWL ontology. In chapter 4, we have described a layered ontology knowledge base architecture based on CIMOn method. However, a feasible and handy API or software tool kit which makes the architecture operational is still lacking. The decision support process is rather a conceptualization and formalization of business work-flow. We have not applied contextual integration in the development of our decision support system prototype, instead, traditional owl:imports was still used to organize ontology modules in the prototype. We think it is worthy to explore how to generalize the architecture and CIMOn method so that they can be applied in general ontology applications. We will address all these limits and issues in our future work.

Bibliography

[1] Franz Baader. *The description logic handbook: theory, implementation, and applications.* Cambridge university press.

[2] Luciano Serafini and Andrei Tamilin. Composing modular ontologies with distributed description logics. In *Modular Ontologies*, pages 321–347. Springer, 2009.

[3] Luciano Serafini and Andrei Tamilin. Composing modular ontologies with distributed description logics. In *Modular Ontologies*, pages 321–347. Springer.

[4] Jie Bao and Vasant G Honavar. Divide and conquer semantic web with modular ontologies: a brief review of modular ontology language formalisms. In *Proceedings of the 1st International Conference on Modular Ontologies-Volume 232*, pages 1–14. CEUR-WS. org, 2006.

[5] Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. 25(1):161–197.

[6] Annie Lefébure and Rafael Rosales Muñoz. Communicating to consumers in Sweden with eco-labels: Is the message getting through? `http://www.diva-portal.org/smash/get/diva2:430238/FULLTEXT02.pdf`. Access date: 2015-08-20.

[7] GEN: Global Ecolabelling Network: What is Ecolabelling? `http://www.globalecolabelling.net/what_is_ecolabelling/index.htm`. Access date: 2015-08-20.

[8] Vilhelmiina Ihamäki, Eva van Ooij, and Stefan van der Panne. Green public procurement in the European Union and the use of eco-labels. `http://ec.europa.eu/environment/gpp/pdf/toolkit/module1_factsheet_ecolabels.pdf`. Access date: 2015-08-20.

[9] Moving towards a circular economy - Environment - European Commission. `http://ec.europa.eu/environment/circular-economy/index_en.htm`. Access date: 2015-08-24.

[10] FlipSnack - Global Ecolabelling Network AGM 2014 Report by Michael Hooper. `http://www.flipsnack.com/FA69B9F569B/global-ecolabelling-network-agm-2014-report.html`. Access date: 2015-08-24.

[11] Global Ecolabel Monitor - Towards Transparency. http://www.ecolabelindex.com/downloads/Global_Ecolabel_Monitor2010.pdf. Access date: 2015-08-24.

[12] 18th European Forum on Eco-innovation: Boosting competitiveness and innovation. http://ec.europa.eu/environment/ecoinnovation2015/1st_forum/presentations_en.html. Access date: 2015-08-25.

[13] GEN31-global Ecolabelling Network magazine with the latest news from organisations promoting environmentally preferable products and services worldwide. http://www.flipsnack.com/FA69B9F569B/gen31.html. Access date: 2015-08-25.

[14] The EU Eco-label scheme - What is it? http://www.ecolabels.fr/en/the-eu-eco-label-scheme-what-is-it.

[15] Facts and Figures - Ecolabel - EUROPA. http://ec.europa.eu/environment/ecolabel/facts-and-figures.html. Access date: 2015-08-21.

[16] How are guidelines drawn up? / the EU eco-label scheme - What is it? http://www.ecolabels.fr/en/the-eu-eco-label-scheme-what-is-it/how-are-guidelines-drawn-up. Access date: 2015-08-21.

[17] Eco-labeling Markets. International Institute for Sustainable Development. http://www.iisd.org/business/markets/eco_labeling.aspx. Access date: 2015-08-19.

[18] Sandra Hoibian. Développement durable et écolabel: état de l'opinion en 2012. http://www.credoc.fr/pdf/Sou/CREDOC_Connaissance_du_Developpement_durable_en_2012.pdf. Access date: 2015-08-21.

[19] Corporate Sustainability Initiative, Jay S. Golden, Dan Vermeer, Bob Clemen, and M. E. M. Davie Nguyen. An overview of ecolabels and sustainability certifications in the global marketplace. 2010.

[20] Sara Olsen and Brett Galimidi. Catalog of approaches to impact measurement: assessing social impact in private ventures. *The Rockfeller Foundation*, 2008.

[21] Mario F. Teisl, Jonathan Rubin, and Caroline L. Noblet. Non-dirty dancing? interactions between eco-labels and consumers. 29(2):140–159.

[22] J. A. Ottman. Industry's response to green consumerism: Journal of business strategy, 13 (4), 3–7 (july/august 1992). 26(1):153.

[23] Global Sustainability Standards Symposium | ISEAL Alliance. http://www.isealalliance.org/about-us/annual-conference/global-sustainability-standards-conference. Access date: 2015-08-25.

[24] Jon Andersson, Björn Johansson, Jonatan Berglund, and Anders Skoogh. Framework for ecolabeling using discrete event simulation. In *Proceedings of the 2012 Symposium on Emerging Applications of M&S in Industry and Academia Symposium*, page 9. Society for Computer Simulation International, 2012.

[25] Swee Leong, Y Tina Lee, and Frank Riddick. A core manufacturing simulation data information model for manufacturing applications. In *Simulation Interoperability Workshop, Simulation Interoperability and Standards Organization*, pages 1–7, 2006.

[26] Raymond Houe and Bernard Grabot. Knowledge modeling for eco-design. *Concurrent Engineering*, 15(1):7–20, 2007.

[27] Raymond Houe and Bernard Grabot. Assessing the compliance of a product with an eco-label: From standards to constraints. *International Journal of Production Economics*, 121(1):21–38, 2009.

[28] Djoko S Sayogo, Weijia Ran, Giri Kumar Tayi, Joanne S Luciano, Luis F Luna-Reyes, Nicolau Depaula, Holly Jarman, Jing Zhang, Jana Hrdinova, Deborah Lines Andersen, et al. Ontological modeling of certification and inspection process to support smart disclosure of product information. *International Journal of Public Administration in the Digital Age (IJPADA)*, 3(2):86–108, 2016.

[29] Sarfraz Ul Haque Minhas and Ulrich Berger. Ontology based environmental knowledge management - a system to support decisions in manufacturing planning. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (IC3K 2014)*, pages 397–404, 2014.

[30] Riichiro Mizoguchi. Part 1: Introduction to ontological engineering. *New Generation Computing*, 21(4):365–384, Dec 2003.

[31] Eva Blomqvist. The use of semantic web technologies for decision support–a survey. *Semantic Web*, 5(3):177–201, 2014.

[32] Virginie Fortineau, Xenia Fiorentini, Thomas Paviot, Ludovic Louis-Sidney, and Samir Lamouri. Expressing formal rules within ontology-based models using swrl: an application to the nuclear industry. *International Journal of Product Lifecycle Management*, 7(1):75–93, 2014.

[33] Artem Katasonov and Marko Palviainen. Towards ontology-driven development of applications for smart environments. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 696–701. IEEE, 2010.

[34] Justin Rockwell, Ian R Grosse, Sundar Krishnamurty, and Jack C Wileden. A decision support ontology for collaborative decision making in engineering design. In *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*, pages 1–9. IEEE, 2009.

[35] Matt-Mouley Bouamrane, Alan Rector, and Martin Hurrell. Development of an ontology for a preoperative risk assessment clinical decision support system. In *Computer-Based Medical Systems, 2009. CBMS 2009. 22nd IEEE International Symposium on*, pages 1–6. IEEE, 2009.

[36] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. 284(5):1–5.

[37] Semantic Web - W3C. `http://www.w3.org/standards/semanticweb/`. Access date: 2015-08-26.

[38] Representing knowledge in the semantic web - slide "the semantic web stack". `http://www.w3c.it/talks/2005/openCulture/slide7-0.html`. Access date: 2015-08-26.

[39] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer.

[40] Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[41] Mark A Musen. Dimensions of knowledge sharing and reuse. *Computers and biomedical research*, 25(5):435–467, 1992.

[42] Natalya F. Noy and Deborah L. McGuinness. *Ontology development 101: A guide to creating your first ontology*. Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880.

[43] Philippe Adjiman, Philippe Chatalic, François Goasdoué, M.-C. Rousset, and Laurent Simon. Somewhere in the semantic web. In *Principles and Practice of Semantic Web Reasoning*, pages 1–16. Springer.

[44] Gilles Falquet, Claudine Métral, Jacques Teller, and Christopher Tweed. *Ontologies in urban development projects*. Springer Science & Business Media, 2011.

[45] Resource description framework (RDF) model and syntax specification. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/#intro`. Access date: 2015-08-27.

[46] RDF schema 1.1. `http://www.w3.org/TR/2014/REC-rdf-schema-20140225/#ch_introduction`. Access date: 2015-08-27.

[47] RDF 1.1 primer. `http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/`. Access date: 2015-08-27.

[48] OWL 2 web ontology language primer. `http://www.w3.org/TR/owl2-primer/`. Access date: 2014-11-30.

[49] OWL 2 web ontology language document overview (second edition). `http://www.w3.org/TR/owl-overview/`. Access date: 2015-08-27.

[50] Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Combining owl ontologies using e-connections. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1):40–59, 2006.

[51] Daniele Nardi and Ronald J. Brachman. An introduction to description logics. In *Description logic handbook*, pages 1–40.

[52] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics. In *Handbook on ontologies*, pages 21–43. Springer.

[53] Franz Baader and Ulrike Sattler. An overview of tableau algorithms for description logics. 69(1):5–40.

[54] York Sure, Steffen Staab, and Rudi Studer. Ontology engineering methodology. In *Handbook on ontologies*, pages 135–152. Springer, 2009.

[55] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[56] H. Sofia Pinto, C. Tempich, and Steffen Staab. Ontology engineering and evolution in a distributed world using DILIGENT. In *Handbook on ontologies*, pages 153–176. Springer.

[57] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer, 2012.

[58] Yue Pan, Guotong Xie, Li Ma, Yang Yang, ZhaoMing Qiu, and Juhnyoung Lee. Model-driven ontology engineering. *Journal on Data Semantics VII*, pages 57–78, 2006.

[59] Paul Doran, Valentina Tamma, and Luigi Iannone. Ontology module extraction for ontology reuse: an ontology engineering perspective. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 61–70. ACM, 2007.

[60] Dmitry Tsarkov. Improved algorithms for module extraction and atomic decomposition. In *Description Logics*, 2012.

[61] Mathieu d'Aquin, Anne Schlicht, Heiner Stuckenschmidt, and Marta Sabou. Criteria and evaluation for ontology modularization techniques. *Modular ontologies*, pages 67–89, 2009.

[62] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Extracting modules from ontologies: A logic-based approach. In *Modular Ontologies*, pages 159–186. Springer, 2009.

[63] Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra. *Modular ontologies: concepts, theories and techniques for knowledge modularization*, volume 5445. Springer.

[64] Yimin Wang, Peter Haase, and Jie Bao. A survey of formalisms for modular ontologies. In *Workshop on Semantic Web for Collaborative Knowledge Acquisition*, 2007.

[65] Alan Rector, Sebastian Brandt, Nick Drummond, Matthew Horridge, Colin Pulestin, and Robert Stevens. Engineering use cases for modular development of ontologies in owl. *Applied Ontology*, 7(2):113–132, 2012.

[66] Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Ontology integration using e-connections. In *Modular ontologies*, pages 293–320. Springer.

[67] Jie Bao, Giora Slutzki, and Vasant Honavar. A semantic importing approach to knowledge reuse from multiple ontologies. In *AAAI*, volume 7, pages 1304–1309, 2007.

[68] Jie Bao, George Voutsadakis, Giora Slutzki, and Vasant Honavar. Package-based description logics. In *Modular Ontologies*, pages 349–371. Springer, 2009.

[69] Jie Bao, Doina Caragea, and Vasant Honavar. A distributed tableau algorithm for package-based description logics. In *the 2nd International Workshop On Context Representation And Reasoning (CRR 2006), co-located with ECAI 2006*.

[70] Ratnesh Sahay, Antoine Zimmermann, Ronan Fox, Axel Polleres, and Manfred Hauswirth. A formal investigation of semantic interoperability of hcls systems. *Healthc Adm: Concepts Methodologies Tools Appl: Concepts Methodologies Tools Appl*, 177, 2014.

[71] Paolo Bouquet, Fausto Giunchiglia, Frank Van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-owl: Contextualizing ontologies. In *International Semantic Web Conference*, volume 2870, pages 164–179. Springer, 2003.

[72] Heiner Stuckenschmidt and Michel Klein. Integrity and change in modular ontologies. In *IJCAI*, pages 900–908, 2003.

[73] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer, 2009.

[74] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1):161–197, 1998.

[75] Jyotishman Pathak, Thomas M Johnson, and Christopher G Chute. Survey of modular ontology techniques and their applications in the biomedical domain. *Integrated computer-aided engineering*, 16(3):225–242, 2009.

[76] Richard J Williams, Neo D Martinez, and Jennifer Golbeck. Ontologies for ecoinformatics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(4):237–242, 2006.

[77] William Greenly, Charles Sandeman-Craik, Yago Otero, and John Streit. Case study: Contextual search for volkswagen and the automotive industry. *Tribal DDB UK*, 2011.

[78] Marcela Vegetti, Horacio Leone, and Gabriela Henning. Pronto: An ontology for comprehensive and consistent representation of product information. *Engineering Applications of Artificial Intelligence*, 24(8):1305–1327, 2011.

[79] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[80] Christine Parent and Stefano Spaccapietra. An overview of modularity. In *Modular Ontologies*, pages 5–23. Springer, 2009.

[81] Aldo Gangemi. Ontology design patterns for semantic web content. In *International semantic web conference*, pages 262–276. Springer, 2005.

[82] Aldo Gangemi, Asunción Gómez-Pérez, Valentina Presutti, and Mari Carmen Suárez-Figueroa. Towards a catalog of owl-based ontology design patterns. 2007.

[83] Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Ontology integration using ε-connections. In *Modular ontologies*, pages 293–320. Springer, 2009.

[84] Luciano Serafini and Andrei Tamilin. Drago: Distributed reasoning architecture for the semantic web. In *The Semantic Web: Research and Applications*, pages 361–376. Springer, 2005.

[85] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(2):93–136, 1996.

[86] Dean Jones, Trevor Bench-Capon, and Pepijn Visser. Methodologies for ontology development. 1998.

[87] Philipp Cimiano and Johanna Völker. A framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), Lecture Notes in Computer Science, Springer*, volume 3513, pages 227–238. Springer, 2005.

[88] Fabien Gandon. *Distributed Artificial Intelligence and Knowledge Management: ontologies and multi-agent systems for a corporate semantic web*. PhD thesis, Université Nice Sophia Antipolis, 2002.

[89] Yimin Wang, York Sure, Robert Stevens, and Alan Rector. Knowledge elicitation plug-in for protégé: Card sorting and laddering. *The Semantic Web–ASWC 2006*, pages 552–565, 2006.

[90] Luciano Serafini and Martin Homola. Modular knowledge representation and reasoning in the semantic web. In *Semantic Web Information Management*, pages 147–181. Springer, 2010.

[91] Janez Brank, Marko Grobelnik, and Dunja Mladenić. A survey of ontology evaluation techniques. 2005.

[92] Da Xu, Mohamed Hedi Karray, and Bernard Archimède. A semantic-based decision support platform to assist products' eco-labeling process. *Industrial Management & Data Systems*, 117(7):1340–1361, 2017.

[93] Denny Vrandečić. Ontology evaluation. In *Handbook on Ontologies*, pages 293–313. Springer, 2009.

[94] Marta Sabou and Miriam Fernandez. Ontology (network) evaluation. In *Ontology engineering in a networked world*, pages 193–212. Springer, 2012.

[95] Bijan Parsia, Ulrike Sattler, and Thomas Schneider. Mechanisms for importing modules. In *Proceedings of the 6th International Conference on OWL: Experiences and Directions-Volume 529*, pages 19–28. CEUR-WS. org, 2009.

[96] Anika Gross, Michael Hartung, Toralf Kirsten, and Erhard Rahm. Mapping composition for matching large life science ontologies. In *ICBO*, pages 109–116, 2011.

[97] Prasenjit Mitra, Gio Wiederhold, and Martin Kersten. A graph-oriented model for articulation of ontology interdependencies. In *International Conference on Extending Database Technology*, pages 86–100. Springer, 2000.

[98] Heiner Stuckenschmidt, Frank Van Harmelen, Luciano Serafini, Paolo Bouquet, and Fausto Giunchiglia. Using c-owl for the alignment and merging of medical ontologies. Technical report, University of Trento, 2004.

[99] Joana Hois, Mehul Bhatt, and Oliver Kutz. Modular ontologies for architectural design. In *FOMI*, pages 66–77, 2009.

[100] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.

[101] Till Mossakowski, Mihai Codescu, Fabian Neuhaus, and Oliver Kutz. The distributed ontology, modeling and specification language–dol. In *The Road to Universal Logic*, pages 489–520. Springer, 2015.

[102] Christoph Lange, Till Mossakowski, and Oliver Kutz. Lola: a modular ontology of logics, languages and translations. 2012.

[103] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment api 4.0. *Semantic web*, 2(1):3–10, 2011.

[104] Jérôme Euzenat, François Scharffe, and Antoine Zimmermann. Expressive alignment language and implementation. 2007.

[105] Jérôme Euzenat, Axel Polleres, and François Scharffe. Processing ontology alignments with sparql. In *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on*, pages 913–917. IEEE, 2008.

[106] Evren Sirin and Bijan Parsia. Sparql-dl: Sparql query for owl-dl. In *OWLED*, volume 258, 2007.

[107] Antoine Zimmermann. Integrated distributed description logics. In *Proc. 20th International workshop on description logic (DL)*, pages 507–514. Bolzano University Press, 2007.

[108] Boontawee Suntisrivaraporn. Module extraction and incremental classification: A pragmatic approach for el+ elˆ+ ontologies. *The Semantic Web: Research and Applications*, pages 230–244, 2008.

[109] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.

[110] Bernardo Cuenca Grau and Oliver Kutz. Modular ontology languages revisited. In *SWeCKa 2007: Proc. of the IJCAI- 2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition ,Hyderabad ,India ,January 7 ,2007*, 2007.

[111] Hanen Abbes and Faiez Gargouri. Structure based modular ontologies composition.

[112] Luciano Serafini, Alexander Borgida, and Andrei Tamilin. Aspects of distributed and modular ontology reasoning. In *IJCAI*, volume 5, pages 570–575, 2005.

[113] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta, and Aldo Gangemi. *Ontology engineering in a networked world*. Springer Science & Business Media, 2012.

[114] Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Modularity and web ontologies. In *KR*, pages 198–209, 2006.

[115] Jay S Golden, Dan Vermeer, Bob Clemen, A Michalko, D Nguyen, C Noyes, et al. An overview of ecolabels and sustainability certifications in the global marketplace. *Corporate sustainability initiative. Durham: Nicholas Institute for Environmental Policy Solutions, Duke University*, 2010.

[116] Roderick G Whyte. What is a decision support system? *Industrial Management & Data Systems*, 86(7/8):28–30, 1986.

[117] Sean B Eom. Decision support systems research: current state and trends. *Industrial Management & Data Systems*, 99(5):213–221, 1999.

[118] Mary Ann Curran, Anastassia Manuilova, Klas Hallberg, Karin Sanne, Johan Widheden, and Malin Bogeskär. Dantes: a demonstration project for sustainable development. *Management of Environmental Quality: An International Journal*, 16(2):143–152, 2005.

[119] Louise Camilla Dreyer, Anne Louise Niemann, and Michael Z Hauschild. Comparison of three different lcia methods: Edip97, cml2001 and eco-indicator 99. *The international journal of life cycle assessment*, 8(4):191–200, 2003.

[120] Sverker Molander, Peter Lidholm, Diederik Schowanek, Mar Recasens, Pere Fullana i Palmer, Frans M Christensen, Jeroen B Guinee, Michael Hauschild, Olivier Jolliet, David W Pennington, et al. Omniitox-operational life-cycle impact assessment models and information tools for practitioners. *The International Journal of Life Cycle Assessment*, 9(5):282–288, 2004.

[121] Olivier Jolliet, Manuele Margni, Raphaël Charles, Sébastien Humbert, Jérôme Payet, Gerald Rebitzer, and Ralph Rosenbaum. Impact 2002+: a new life cycle impact assessment methodology. *The International Journal of Life Cycle Assessment*, 8(6):324–330, 2003.

[122] Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In *International Joint Conference on Automated Reasoning*, pages 292–297. Springer, 2006.

[123] Rob Shearer, Boris Motik, and Ian Horrocks. Hermit: A highly-efficient owl reasoner. In *OWLED*, volume 432, page 91, 2008.

[124] Bertrand Meyer. *Object-oriented software construction*, volume 2. Prentice hall New York, 1988.

[125] Luigi Iannone, Ignazio Palmisano, Alan L Rector, and Robert Stevens. Assessing the safety of knowledge patterns in owl ontologies. In *Extended Semantic Web Conference*, pages 137–151. Springer, 2010.